COMPUTATIONAL DESIGN OF MICROVASCULAR BIOMIMETIC MATERIALS

BY

ALEJANDRO MARCOS ARAGÓN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Doctoral Committee:

Professor C. Armando Duarte, Chair
Professor Philippe H. Geubelle
Professor Daniel A. Tortorelli
Professor Scott R. White

Abstract

Biomimetic microvascular materials are increasingly considered for a variety of autonomic healing, cooling and sensing applications. The microvascular material of interest in this work consists of a network of hollow microchannels, with diameters as small as 10 μm, embedded in a polymeric matrix. Recent advances in the manufacturing of this new class of materials have allowed for the creation of very complex 2D and 3D structures. The computational design of such network structures, which is the focus of this work, involves a set of particular challenges, including a large number of design variables (e.g., topology of the network, number of diameters to consider and their sizes) that define the network, and a large number of multidisciplinary objective functions and constraints that drive the optimization process. The computational design tool to be developed must be capable of capturing the trade-off between the different objective and constraint functions, as, for example, networks designed for flow efficiency are likely to have a topology that is very different from those designed for structural integrity or thermal control.

In this work, we propose to design these materials using Genetic Algorithms (GAs), the most common methodology within a broader category of Evolutionary Algorithms (EAs). GAs can be combined with a Pareto-selection mechanism to create Multi-Objective Genetic Algorithms (MOGAs), which enable the optimization of an arbitrary number of objective functions. As a result, a Pareto-optimal front is obtained, where all candidates are optimal solutions to the optimization problem. Adding a procedure to deal with constraints results in a powerful tool for multi-objective constrained optimization. The method allows the use of discrete variable problems and it does not require any a priori knowledge of the optimal solution. Furthermore, GAs search the entire decision space so the optimal solutions found are likely to be global. The MOGA optimization framework is also combined with a physical solver based on advanced finite element methods to study the thermal behavior of these materials. Because the MOGA requires a vast number of individual evaluations, emphasis is placed on the computational efficiency of the solver. Thus, a simplified formulation is used to take into account the cooling effect of the fluid, instead of solving the conjugate heat transfer problem for obtaining the temperature field in both solid and fluid domains. The Generalized Finite Element Method (GFEM) is adopted because accurate finite element approximations of the temperature field can be

obtained on finite element meshes that are independent of the geometry of the embedded network. Numerical experiments of multi-physics optimization involving flow efficiency, void volume fraction and thermal control are presented. Results show that the trade-offs between conflicting objectives is well captured so that the optimal design is readily available to the analyst.

To my mother, for her unconditional love and support, and for always pushing me to give the best in every situation – thank you.

To the memory of Dr. Jeronymo Peixoto. A. Pereira, who will always be remembered as a great mind, and an excellent friend.

Acknowledgments

This work would not have been possible without the aid of several people and organizations. First, I am grateful to my mother, for believing in my dreams and supporting my decisions, for constantly pushing me for excellence and for always being present when we were thousands of kilometers away from each other. To my uncle, for being a source of inspiration since a very young age. To my brother Fernando Aragón and my nephews Fernando and Gabriela Koester, for being a great source of joy in my life. I am beholden to Maryline Penedo, who was by my side and supported me throughout most of these years. I am indebted to my advisor, Philippe H. Geubelle, for his constant guidance and support throughout my Ph.D., and for making it a wonderful experience. To my undergraduate students Jessica Wayer, Kyle Smith, and Edmond Helaire, for contributing to this work under the Undergraduate Research Opportunity Program (UROP), sponsored by the NASA's Illinois Space Consortium. I am greatly thankful to Prof. C. Armando Duarte, for his collaboration and innumerable discussions. I would like to thank Prof. Scott White, for providing guidance and for being a terrific boss. I am also grateful to Prof. Daniel A. Tortorelli, for accepting to be a member of my doctoral committee and for providing valuable insight during the preliminary examination. Many thanks to my colleagues John Fettig, Mohan Kulkarni, Karthik Srinivasan, Helen Inglis, Fernando Stump, Premsainath Selvarasu and Soheil Soghrati, for the countless debates and fruitful exchange of ideas. I would also like to acknowledge my fellow colleagues in the Autonomous Materials Systems group at the University of Illinois, Christopher Hansen, Willie Wu, Brian Kozola, Rajat Saksena and Vinay Natrajan, for collaborating with me from the *dark side of the Force* (i.e., the experiments). Many thanks to my Karate Sensei-tachi Robert Dignan, Jeremy Epstein, Eric Senger, Dan Rubel, Kelsey Grandi, Matt Grobis, and Albert Liao, and my other friends in the Illinois Shotokan Karate Club at the University of Illinois. For showing me the marvelous world of Tango, I am greatly thankful to Chantelle Hougland and Tajal Patel, To my good friends Harkanwal Singh Brar, Shweta Sharma, Alessandro Bellina, Kevin Farrell, Daniela Lindner, Paula Mendez Keil, Zhanar Abil, Igor Cunha, Cristian Gaedicke, Raul A. Mora, Ricardo Mejia-Alvarez, Diego Margot, Pablo Acosta, Yovanni Cataño, Alexander Kunzelman, Eun Mónica Lim, Soheil Soghrati, Daniel Daza, Jagadish Kumar, Miguel Salazar Franco, Evangelos Benos, Victor Nyekel, and Vasilis Paschalidis,

for making me feel as if I were at home. I am grateful to the Fulbright program, which provided me with support during the Master of Science program and without which any of this would not be possible. Finally, this work would not have been possible without the financial support provided by the AFOSR (MURI Grant No. F49550-05-1-0346).

Table of Contents

List of tables

List of figures

List of abbreviations

μ−PIV	Microscopic Particle Image Velocimetry
AFOSR	Air Force Office of Scientific Research
BGL	Boost Graph Library
BLAS	Basic Linear Algebra Subprograms
CPU	Central Processing Unit
CRTP	Curiously Recurring Template Pattern
DC	Deterministic Crowding
EA	Evolutionary Algorithm
FEM	Finite Element Method
GA	Genetic Algorithm
GDE	Generalized Differential Evolution
GFEM	Generalized FEM
GNU	GNU's Not Unix!
I/O	Input/Output
IR	Infrared
LIF	Laser Induced Fluorescence
MOEA	Multi-Objective Evolutionary Algorithms
MOGA	Multi-Objective Genetic Algorithm
MOOP	Multi-Objective Optimization Problems
MPI	Message Passing Interface
MTL	Matrix Template Library
MURI	Multidisciplinary University Research Initiative

NSGA-II	Non-dominated Sorting Genetic Algorithm II
SGA	Simple Genetic Algorithm
URL	Uniform Resource Locator
VTK	Visualization Toolkit
XFEM	eXtended FEM

List of symbols

English letter symbols

A_f	Area of the cross section of a fin specimen, [m^2]
A_Ω	Area of a polymeric component, [m^2]
$\mathcal{A}$	Allele of a chromosome
a	First dimension of a 2D rectangular domain or a 3D cuboid
b	Second dimension of a 2D rectangular domain or a 3D cuboid
C^0	Set of continuous functions with discontinuous first derivatives
$\mathcal{C}$	Chromosome of an individual
c	Ramp correction function
c	Third dimension of a 3D cuboid
c_p	Fluid specific heat capacity, [J/kg K]
D_e	Microchannel diameter, [μm]
D_i	$i-$th channel diameter in the set $\mathcal{D}$, [μm]
$\mathcal{D}$	Diameter set used in the optimization process
d	Distance function to a line segment
$\bar{d}$	Required minimum degree of a network
E_p	Efficiency
$\mathcal{E}, \mathcal{E}'$	Sets of edges in a graph, $\mathcal{E}' \subseteq \mathcal{E}$
$\mathbb{E}^N$	$N-$dimensional Euclidean space
$\mathbf{e}_i$	$i-$th basis vector of the Euclidean space
$\mathbf{e}_{\xi_e}$	Unit vector a long the microchannel axial coordinate

$\mathbf{e}_{\eta_e}$	Unit vector a long the microchannel radial coordinate
e, e'	Graph edge, $e \in \mathcal{E}$, $e' \in \mathcal{E}'$
F_i	$i-$th non-dominated front
f	Heat source, [W/m^3]
$\mathcal{G}, \mathcal{G}'$	Graphs, $\mathcal{G}' \subseteq \mathcal{G}$
$\mathcal{G}_i$	$i-$th sub-domain boundary of domain Ω
H	Hilbert space
h	Heat transfer coefficient, [W/m^{2}K]
h_f	Film coefficient for a fin, [W/m^{2}K]
h_α	Scaling parameter related to the size of the cloud of node $\mathbf{x}_\alpha$
$\mathcal{I}$	Individual of a population
$\vec{K}$	Network characteristic matrix
k	Chromosome cardinality
L_e	Microchannel length, [m]
$L_{\alpha i}$	$i-$th enrichment function associated with node $\mathbf{x}_\alpha$
$\mathcal{L}$	Enrichment function set
M	Number of finite elements in discretization Ω^h
M	Möes ridge enrichment function
$\dot{m}_e$	Microchannel mass flow rate, [kg/s]
$\dot{m}_v$	Net mass flow rate leaving or entering vertex v, [kg/s]
$\dot{m}_{\mathcal{S}}$	Total network mass inflow rate, [kg/s]
$\dot{m}_{\mathcal{T}}$	Total network mass outflow rate, [kg/s]
$\vec{\dot{m}}$	Network mass flow rate vector
N	Number of finite element nodes in discretization Ω^h
n	Number of vertex locations along a Cartesian direction
P	Power, [W]
P_f	Perimeter of the cross section of a fin specimen, [m]

$\mathcal{P}$	Population of a genetic algorithm
$\mathcal{P}_0$	Initial population of a genetic algorithm
$\mathcal{P}^u$	Set of unfeasible individuals within a population $\mathcal{P}$
$\mathcal{P}^f$	Set of feasible individuals within a population $\mathcal{P}$
p	Pressure, [Pa]
$\vec{p}$	Network pressure vector
p_c	Probability of crossover
p_m	Probability of mutation
$\bar{q}$	Prescribed heat flux, [W/m^2]
R	Junction ramp enrichment
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^N$	$N-$dimensional real coordinate space
Re	Reynolds number
$\mathbf{r}$	Redundancy matrix
r	Ramp enrichment function
r_{ij}	Number of node disjoints paths between vertices v_i and v_j
$\bar{r}$	Network average redundancy
S_p	Speed-up
$\mathcal{S}$	Set of vertices representing inflows
s	Tournament selection pressure
T_s	Sequential computation time, [s]
T_p	Parallel computation time, [s]
$\mathcal{T}$	Set of vertices representing outflows
t	GA generation
$t_{\max}$	GA maximum number of generations
t_c	Communication time, [s]
$\bar{t}_e$	Average evaluation time, [s]

$\mathcal{U}$	Set of trial functions
u	Temperature, [K]
$\bar{u}$	Prescribed temperature, [K]
u_∞	Ambient temperature, [K]
V_Ω	Volume of a polymeric component, [m^3]
$\mathcal{V}$, $\mathcal{V}''$	Sets of vertices in a graph, $\mathcal{V}'' \subseteq \mathcal{V}$
$\mathcal{V}$	Set of weight functions
v	Network void volume fraction
$\mathbf{v}_e$	Microchannel fluid velocity field, [m/s]
v, v'	Graph vertex, $v \in \mathcal{V}$, $v' \in \mathcal{V}''$
$\mathbf{x}$	Space coordinate
$\mathbf{x}_J$	Junction space coordinate
$\mathbf{x}_\alpha$	Space coordinate associated with node α in a finite element mesh
$\mathbb{Z}$	Set of natural numbers
z	Natural number, $z \in \mathbb{Z}$

Greek letter symbols

Γ	Boundary of the domain Ω
Γ_u	Part of the boundary with prescribed temperature boundary conditions
Γ_q	Part of the boundary with prescribed heat flux boundary conditions
Γ_h	Part of the boundary with prescribed convective boundary conditions
γ	Ratio between average evaluation and communication times
Δp	Pressure drop, [Pa]
δ	Cartesian grid spacing, [m]
$\delta(\mathcal{G})$	Minimum degree of the vertices of graph $\mathcal{G}$
ε	Perturbation parameter used in the semi-positive definite linear system solver

ζ Flow homogeneity parameter

η Node normalized y coordinate

η_e Microchannel radial coordinate, [m]

η_i^D Counter used in the degree constraint ψ_D

η_i^F Counter used in the non-zero flow constraint ψ_F

η_{ij}^I Counter used in the intersection constraint ψ_I

$\boldsymbol{\kappa}$ Thermal conductivity tensor

κ Thermal conductivity, [W/m K]

Λ Point lattice

Λ_j^i Level set function value with respect to the $i-$th line segment at the $j-$th node

ν Fluid kinematic viscosity, [m^2/s]

ξ Node normalized x coordinate

ξ_e Microchannel axial coordinate, [m]

ξ_h Hydrodynamic entry length, [m]

ρ_f Fluid density, [kg/m^3]

Φ Set of objective functions $\left\{\phi_i\right\}$ that define the objective function space

$\boldsymbol{\phi}$ Vector of objective functions

ϕ_f Flow efficiency objective function, [Pa]

ϕ_h Flow homogeneity objective function

ϕ_s Surface convected energy objective function, [W/m^2]

ϕ_T Maximum temperature objective function, [K]

ϕ_v Void volume fraction objective function

φ_α Lagrangian shape function associated with node $\mathbf{x}_\alpha$

Ψ Set of constraint functions $\{\psi_i\}$ that define the constraint function space

$\boldsymbol{\psi}$ Vector of constraints

ψ_c Connectivity constraint

ψ_D Degree constraint

ψ_F	Non-zero flow constraint
ψ_I	Intersection constraint
$\psi_{\alpha i}$	Resulting enrichment function used at node $\mathbf{x}_\alpha$
Ω	Domain of the microvascular material

Superscripts

0	It refers to quantities associated with the microvascular network template
h	Refers to a discretized quantity
N	Problem dimensionality

Subscripts

e, e'	Refers to a graph edge
f	Fluid
h	Refers to a prescribed convective boundary condition
p	Parallel
q	Refers to a prescribed heat flux boundary condition
ref	Refers to the reference configuration used to normalize objective function values
s	Sequential
u	Refers to a prescribed temperature boundary condition
v, v'	Refers to a graph vertex
α	Refers to a node in a finite element mesh
Φ	Refers to the objective function space
Ψ	Refers to the constraint function space

Other

$(\cdot, \cdot)$	Linear operator
$a(\cdot, \cdot)$	Bilinear operator
$\deg(v)$	Degree of a vertex v
$\mathcal{H}(\cdot)$	Heaviside step function
$\delta(\cdot)$	Dirac delta function
$\|\cdot\|_{L_2(\Omega)}$	L_2 norm
$\|\cdot\|_{E(\Omega)}$	Energy norm
$[\![\cdot]\!]$	Jump of a field
$\ll_\Psi$	Constraint space domination operator
$\ll_\Phi$	Objective function space domination operator
$\lll$	Global domination operator
$\|\cdot\|$	Number of elements in a set
∇	Vector differential operator
$O(\cdot)$	Asymptotic notation used in the analysis of algorithms (i.e., big-O notation)

1 Introduction

Throughout history, humans have found themselves inspired by nature to create materials, structures, and processes [1, 2]. Without any doubt, a major achievement borrowed from nature is the ability to fly, with the first successful attempt of flying occurring around A.D. 875 in the Caliphate of Córdoba (currently part of Spain) by Abū'l-Qāsim 'Abbās b. Firnās. Inspired by birds, Ibn Firnās attached a couple of wings, covered himself with feathers, and flung himself into the air covering a considerable distance, according to the Moroccan historian al-Maqqarī [3]. Later in the 11^{th} century, Eilmer of Malmesbury is considered the first occidental to have flown [3]. A monk in the Malmesbury Abbey Benedictine monastery, in Wiltshire England, Eilmer attached wings to his arms and legs and flew for more than 180 m in the first decade of A.D. 1000. Later in the early 16^{th}, Leonardo da Vinci's *Codex on the Flight of Birds* details the plans for several flying machines inspired by birds [4]. Another great achievement, inspired by silk, sinew and some plant fibres, was the fabrication of single-polymer materials such as nylon [5]. Other processes borrowed from nature, as listed in [6], include: *i) bacteria:* biological motor; *ii) plants:* chemical energy conservation, superhydrophobicity, self-cleaning, drag reduction, hydrophilicity, adhesion, and motion; *iii) insects, spiders, lizards and frogs:* superhydrophobicity, reversible adhesion in dry and wet environments; *iv) aquatic animals:* low hydrodynamic drag and energy production; *v) birds:* light coloration, camouflage and insulation; *vi) seashells, bones, teeth:* high mechanical strength; *vii) spider web:* biological self-assembly; *viii) moth-eye effect:* antireflective surfaces; *ix) fur and skin of polar bear:* thermal insulation; and *x) biological systems:* self-healing and sensory-aid devices.

Within the category of biological systems, a successful attempt for the creation of materials with autonomous healing consists in embedding microcapsules containing a healing agent that is released in the presence of a crack [7]. The healing agent polymerizes on the surface of the crack after being in contact with catalyst particles that are distributed in the matrix of the polymer. Because of the finite supply of the healing agent contained in the capsules, only a few healing cycles can be achieved and a different approach needs to be undertaken to provide a continuous supply of the agent over time. Self-healing capabilities can also be achieved by incorporating a hollow microvascular network in a polymeric matrix [8, 9], as envisioned in [10]. Advances in the manufacturing of this new

class of materials has allowed for the creation of very complex 2D and 3D structures, with diameters that can be as small as 10 μm [10]. A fully automated nozzle with 3D motion is used to extrude a fugitive ink on a solid polymeric substrate. The scaffold that results from the writing process is then submerged in liquid polymer, and a curing process is started to obtain the bulk of the polymeric component. The ink is then removed from the component by changing its temperature, leaving voids in the the material that constitute the microvascular network. Traditionally, the writing process of the microvascular network is done in 2D, thus the creation of 3D scaffolds is carried out by superimposing 2D layers. However, new methodologies are being explored where a 3D structure is extruded within a gel that serves as a support, as illustrated in Figure 1.1. Even though this technique is at a premature stage, it promises the creation of truly complex 3D microvascular topologies.

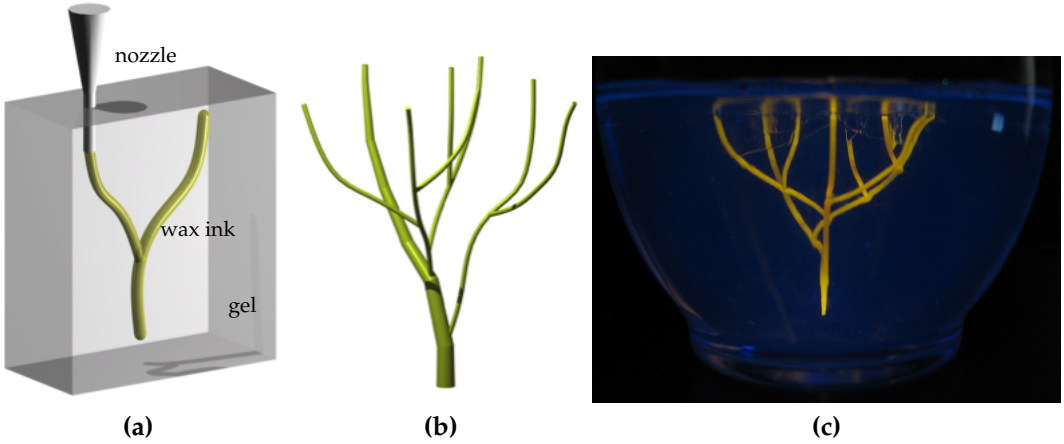


Figure 1.1 – New technologies used in the manufacturing process of 3D microvascular materials. Courtesy of Willie Wu from the Materials Science and Engineering Department at the University of Illinois. (a) Manufacturing process, consisting of a nozzle with 3D motion that extrudes wax ink in a gel. The gel supports the structure so that very complex 3D structures can be created. (b) Prototype structure. (c) Final manufactured sample.

By mimicking vascular systems found in nature, e.g., those found in vascular plants and animals, the use of a microvascular network to provide a healing agent guarantees a continuous supply and thus circumvents the drawback of embedding microcapsules. The resulting functionality of the microvascular polymer depends on the properties of the fluid that flows through the embedded network. These materials can also be used for thermal regulation if a coolant is used, a functionality that was foreseen in [8]. This dissertation focuses on the design of microvascular materials for active cooling applications, where

the polymeric component is subjected to thermal loads. The computational design of the microvascular network embedded in these polymers is an intricate process, for there is a large number of design variables (e.g., the topology of the network, number of diameters and their sizes, dimensionality of the polymeric component) and various desired features. The computational design tool to be developed must be capable of capturing the trade-off between the competing objectives, as for example, networks designed for flow efficiency are likely to have a topology that is very different from those designed for structural integrity or thermal control. Because of the computationally intensive nature of the optimization methodology, we are not only interested in the optimized network topologies, but also in obtaining an efficient and accurate evaluation of the objective functions and constraints involved in the optimization.

Several approaches can be used to design the microvascular networks embedded in these materials. These include *i)* semi-analytical approaches; *ii)* gradient-based search methods; and *iii)* evolutionary algorithms. Semi-analytical approaches can be used to solve network optimization problems that do not involve a large set of optimization variables. These methods often require *a priori* information about the optimal solution. Constructal theory [11] lies in this group, and it has been used for the solution of simple problems [12, 13, 14]. Within gradient-based search techniques, topology optimization has been used recently to optimize flow networks [15]. This method involves design variables that are continuous and sometimes leads to an optimal solution that is local, so finding a global optimal solution is not guaranteed. Borrowing some concepts from biological evolution, Evolutionary Algorithms (EAs) are population-based algorithms that improve a set of candidate solutions throughout their evolution. Genetic Algorithms (GAs) [16, 17] are the most known methodology within this group, and they use *selection*, *crossover* and *mutation* as genetic operators to obtain better candidate solutions. The *survival of the fittest* takes place through selection, preventing the worst candidates to be nominated for reproduction. The distinction among individuals is conducted through the computation of one or more objective functions, which quantitatively describe how fit the individuals are to solve the problem. Through the crossover operator, new candidate solutions are created by mixing the information of those nominated through the selection process. Thus, newly created individuals contain features of their parents. The mutation operator is a small modification of the features of an individual and could introduce new features to the existing solutions. GAs offer several advantages compared to the methods listed above. GAs can use discrete and/or continuous variables, and they search the entire decision space, so optimal solutions found are likely to be global optima. Furthermore, there is no need for *a priori* knowledge about the optimal solutions, but this information can be used

to obtain a starting population with those features.

The Simple Genetic Algorithm (SGA) [16] is probably the most known algorithm in this group and is used with a single objective function, usually determined as a linear combination of weighted objectives if the optimization is on several objectives. In Multi-Objective Genetic Algorithms (MOGAs), individuals are compared in all objective functions using a Pareto-optimality procedure. In this context, an individual is better than another if it is better in at least one objective function and it is not worse in the rest. This selection mechanism, that has been borrowed from the field of economics, results in a powerful method for obtaining an optimal front (called Pareto-optimal), where all individuals lying on it are optimal solutions to the multi-objective optimization problem. By supplementing the MOGA with a mechanism that deals with constraints, a powerful methodology for multiple-objective constrained optimization is obtained. This is the method chosen in our work because it allows us to visualize the trade-offs between conflicting objective functions.

The GA-based optimization tool can be used to study multi-physics optimization problems if the objective functions involved describe different physical phenomena. In the context of active cooling applications, the temperature field is obtained when the polymeric component is subjected to thermal loads. To obtain the temperature distribution in the microvascular material, one needs to solve for the temperature field in both the solid and the fluid domains. This problem is referred to in the literature as the *conjugate heat transfer problem*. However, this approach is not straightforward because of the coupling that exists between both domains if a non-monolitic solution approach is used. This means that an iterative approach is needed to obtain a converged value of the temperature at solid-fluid boundaries. Because the MOGA requires thousands of individual evaluations, and the total CPU time is directly proportional to the cost of individual objective functions, we resort to a simplified formulation that describes the convective heat transfer in the microchannels. This allows us to collapse the cooling behavior of microchannels to heat sinks over lines and solve for the heat equation in the solid only. Therefore, the temperature distribution in the solid domain will have a discontinuous gradient along the direction perpendicular to the microchannels. We use this *a priori* information to develop Generalized Finite Element Method (GFEM) enrichment functions. This technique is adopted here because it allows us to obtain an accurate finite element approximation of the temperature field on finite element meshes that are independent of the geometry of the embedded network. The mesh independence is the main motivation for the use of this technique because the template structure that we use to start the optimization process can have an arbitrary configuration. Furthermore, meshing techniques are less robust in

3D and we plan to extend this research to include such domains.

Most of the chapters in this dissertation are self-contained articles, with their own introduction, literature review, conclusions and so forth. The dissertation is organized as follows: Chapter 2 presents the article entitled "Design of microvascular flow networks using multi-objective genetic algorithms", published in 2008 in the journal *Computer Methods for Applied Mechanics and Engineering* [18]. The chapter presents a brief introduction to GAs and outlines the Nondominated Sorting Genetic Algorithm II (NSGA-II), the multi-objective GA used in this work. The chapter then presents a thorough study on the use of the NSGA-II in the optimization of microvascular materials. The optimization of the topology of the network is carried out considering mainly the void volume fraction left by the microvascular network and its flow efficiency as objective functions, constraining the design to networks that cover entirely the polymeric component. The methodology described in Chapter 2 presents a powerful framework for multi-objective constrained optimization. The framework is independent of the dimensionality of the problem, it can accommodate an arbitrary number of objective functions and constraints, and it is able to capture the trade-offs between conflicting objective functions by providing Pareto-optimal fronts from which the selection of a final network topology is readily available to the analyst. It also respects the imposed constraints by design, therefore eliminating unfeasible individuals from the evolution after a few generations.

Chapter 3 presents the development of enrichment functions for the GFEM to study discontinuous gradient field problems. This work, entitled "Generalized finite element enrichment functions for discontinuous gradient fields", was published in 2009 in the *International Journal for Numerical Methods in Engineering* [19]. A thorough study of enrichment functions available in the literature, as well as new proposed enrichment functions, shows that it is possible to recover optimal convergence rates for this type of problems on finite element meshes that are completely independent of the topology of the microvascular networks. Moreover, it is shown that the use of the GFEM provides more accurate solutions for the same number of degrees of freedom. It is also shown that higher order approximations, obtained by enriching also with polynomial enrichments, do not require any correction to the method to achieve optimal convergence rates. The resulting PDE solver is robust, accurate, and can be used when the microvascular network template follows a strange pattern for which the creation of a matching mesh is not possible.

Our current efforts on the multi-physics optimization of microvascular materials are summarized in Chapter 4. The coupling of the multi-objective constrained optimization framework described in Chapter 2 with the physics solver based on advanced finite element methods described in Chapter 3, provides a powerful methodology for multi-

physics optimization. Thus, this chapter presents the use of these tools and extends the optimization of 2D microvascular networks in the presence of thermal loads. As a result, the topology of the microvascular networks can be designed within the context of active cooling applications. Through several examples, it is shown that well-defined Pareto-optimal fronts are obtained for void volume fraction, flow efficiency, maximum temperature and surface convected energy objective functions.

Chapter 5 extends the methodology presented in Chapter 4 to 3D networks, thus closing the gap between the computational approach and the 3D nature of these materials. A 3D mathematical model is able to capture the heat conduction through the thickness of the specimens, thus providing a more realistic computational model. Validation studies are also presented, and optimal designs are manufactured and tested to validate the design framework. Some concluding remarks and possible areas for further research are given in Chapter 6. Finally, Appendix A discusses some of the implementation details of the computational tools developed for this work.

2 GA-based optimization

This chapter presents the article published in 2008 in the journal *Computer Methods for Applied Mechanics and Engineering* [18].

Design of microvascular flow networks using multi-objective genetic algorithms

Abstract

A multi-objective genetic algorithm is used to design 2D and 3D microvascular networks embedded in bio-mimetic self-healing/active-cooling polymeric materials. A variety of objective functions and constraints are considered, ranging from flow efficiency and homogeneity to network redundancy and void volume fraction. The design variables include the network topology defined over a template, and the microchannel diameters chosen among a finite set of values. The effect of network redundancy, template geometry and microchannel diameters on the Pareto-optimal fronts generated by the genetic algorithm is investigated.

2.1 Introduction

Inspired by vascular networks in living organisms, materials consisting of a network of microchannels embedded in a polymeric matrix offer great potential in various autonomic healing, cooling and sensing applications. One example of a bio-mimetic material uses hollow fibers embedded in a polymeric matrix and filled with an uncured healing agent, which is released when the fibers experience damage [20, 21]. In a recent publication [8], Toohey and co-workers have demonstrated repeated healing of a polymer coating with the aid of a subsurface microvascular network containing a healing agent in monomeric form. As cracks form in the coating, the healing agent is wicked to the crack surfaces through capillarity and encounters solid catalyst particles contained in the coating. The healing process is thus initiated and can be repeated as long as the three-dimensional

microvascular network contained in the polymeric substrate provides enough healing agent to the coating. The circulation of a liquid in the microvascular network is also being considered for thermal management of a structural component subjected to external thermal loading. An example of such an application can be found in [22] for the case of a microvascular network embedded in an epoxy matrix.

Interest in this class of bio-mimetic materials has also been driven by recent advances in manufacturing techniques such as the robotic deposition process, which allows for the creation of complex two- and three-dimensional microvascular network structures [10, 23]. In this process, the microvascular network is drawn by extruding a fugitive ink on a polymer substrate through needle tips so that the resulting microchannels have diameters that can be as small as $10\,\mu$m. After the drawing process has taken place, the resulting structure is embedded in a liquid polymer that is subsequently cured. The ink contained within the polymer is then evacuated by heating the material, leaving the embedded microvascular network. The use of needle tips with different sizes and a fully automated nozzle with three-dimensional motion results in a powerful manufacturing methodology for the creation of complex patterns.

Various methods have been proposed in the literature for the design of flow networks, which presents a set of unique challenges in terms of the complexity of the objective functions, design variables and constraints. In the constructal theory [11], optimal flow structures are obtained by applying the constructal law, which states that the optimal flow structures should provide easier access to the flow than those that are non-optimal. This method has been used to optimize very simple geometries [12, 13, 14]. Another approach relies on topology optimization, which has been used primarily in structural design, but has been recently extended to the design of flow networks by considering the network either as a continuum or as a discrete system [24]. A comprehensive study of flow networks using a discrete topology is given in [15], where the diameter variables are chosen from the positive real number set. Flow networks can also be designed by evolutionary algorithms, a family of biology-inspired methods that are increasingly gaining popularity because of their simplicity and applicability to the optimization of a large set of problems in numerous fields. Starting from a population of candidate solutions, evolutionary algorithms search for better candidates by applying a set of genetic operators and using a function to differentiate those solutions that are more fit to solve the problem. The last twenty years have seen substantial progress in the understanding of the mechanics behind the optimization based on Genetic Algorithms (GAs) [17], the most popular of these evolutionary methods. GAs have been applied successfully to the optimization of flow networks over a wide range of objective functions [25, 26, 27, 28].

The present study considers a discrete topology for the microchannels of the network and selects their diameters from a discrete set of values, in accordance to available tip sizes in the direct printing process mentioned earlier. For the optimization problems investigated in this work, a multi-objective genetic algorithm scheme is adopted for several reasons. First, the adopted GA can readily handle any number of objective functions in a unified formulation and accommodate the discrete nature of the design variables considered in the present study. Second, gradient-based search techniques may converge to a local optimum, while GAs search a much broader portion of the decision space and are thus more likely to yield a global optimum. Third, GAs can readily be hybridized with other search techniques [29], thus combining the robustness of the global evolutionary search with the accuracy of problem-specific local search methods. Finally, specific features of good solutions found in the GA optimization can be used to initialize another GA optimization (knowledge-based GAs).

The design of a microvascular network is application specific, so networks designed for flow efficiency may look very different from those designed for structural performance. Furthermore, when considering multiple objectives, there is a need to capture the trade-off between those objectives in the form of an optimal front from which the network design can be readily selected. This optimal front must be extracted through analytical or computational means, by decomposing the problem into a set of conflicting objectives and constraints. This particular study focuses on the optimization of two- and three-dimensional simple geometries used in the manufacturing of bio-mimetic polymers. Whether the resulting polymer is used in self-healing or active-cooling applications, a fluid will be driven through the microchannels and flow efficiency is thus a key design objective. However, an optimal solution for the flow efficiency typically maximizes the void volume left by the network, so both objectives conflict with one another. Beyond these two primary objective functions, this paper also introduces objectives and constraints associated with flow homogeneity and network redundancy.

Section 2.2 introduces the problem to be solved and the formulation used to obtain the objective functions. Section 2.3 gives a brief introduction to multi-objective Genetic Algorithms. The results are presented in Section 2.4 and Section 2.5 summarizes the conclusions obtained.

2.2 Problem description

The adopted mathematical model of a bio-mimetic material containing a microvascular network consists of two parts, as illustrated in Figure 2.1 for a two-dimensional (2D)

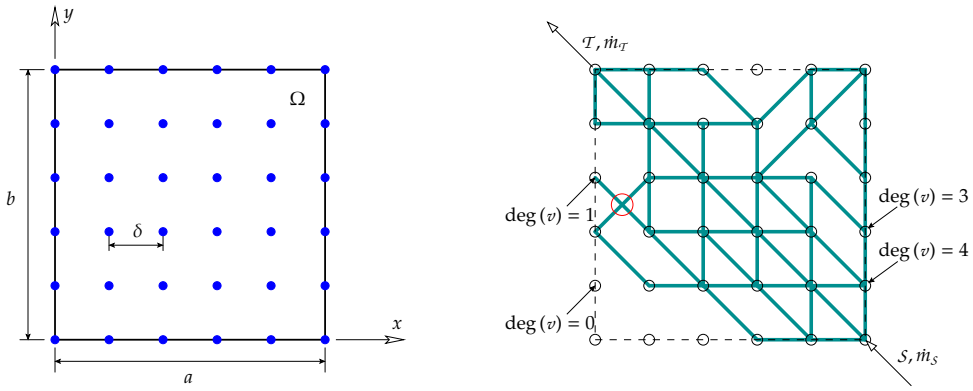


Figure 2.1 – Bio-mimetic material mathematical model. The left figure shows the domain Ω and the 2D point lattice spanned over Cartesian coordinates. The graph data structure that represents the microvascular network is shown on the right. Vertices of the graph laying on top of the 2D point lattice, marked as circles, represent possible microchannel endpoints whereas edges of the graph represent the actual microchannels. Examples of flow boundary conditions (inflow $\mathcal{S}$ and outflow $\mathcal{T}$ locations), together with examples of vertex degrees are also included.

network. First, the bulk of the material is represented by a domain Ω (left Figure 2.1) with area A_Ω (or volume V_Ω). Second, a graph data structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (right Figure 2.1) is used to represent the microvascular network, where each edge e in the set of edges $\mathcal{E}$ of the graph serves as a microchannel and each vertex v in the set of vertices $\mathcal{V}$ as a possible location for microchannel endpoints. Depending on the context, the number of elements in the vertex and edge sets are also denoted by $\mathcal{V}$ and $\mathcal{E}$, respectively. The use of a graph data structure to represent the microvascular network allows us to take advantage of tools associated with graph theory [30]. A graph is *disconnected* if all vertices are not reachable from each other through the existing edges. Another important concept borrowed from graph theory is the *degree* of a vertex, $\deg(v)$, defined as its number of incident edges. This parameter is used to constrain the optimization in order to ensure complete network coverage of the domain and also to investigate the reliability of the network when subjected to damage, as explained in Section 2.2.3.

The current study focuses on convex domain geometries, namely a rectangle in 2D and a cuboid in 3D. We denote by a and b the sides of a rectangular 2D domain and by a, b and c the lengths of the edges of a 3D cuboid. Since there is an infinite number of possible

network configurations that fit within the domain boundary, we restrict the number of possible vertex locations to a set of points located over an $N-$dimensional lattice,

$$\Lambda := \left\{ \sum_{i=1}^{N} z_i \mathbf{e}_i \quad | \quad z_i \in \mathbb{Z} \right\}, \qquad \Lambda \subset \Omega,$$

where the basis vectors for the Euclidian space $\mathbb{E}^N$ have equal magnitude in all directions $\|\mathbf{e}_i\| = \delta$. In other words, the possible vertex locations are limited to the intersection vertices of a regular Cartesian grid having grid spacing δ, such that two neighboring vertices on any Cartesian axis direction are separated by a constant distance. The vertices of the graph data structure corresponding to the point lattice are shown as open circles in the right figure. Similarly, the number of possible edges is limited to those connecting neighboring vertices. With these simplifications, the number of possible network configurations is drastically reduced. Several properties are assigned to the graph data structure that represents the microvascular network. A coordinate property in $\mathbb{E}^N$ is associated with each vertex corresponding to a point in the lattice. Each edge e represents a cylindrical microchannel and has a length L_e and diameter D_e. Networks containing microchannels oriented only in the directions of the coordinate axes will be henceforth referred to as *orthogonal*. Similarly, we will refer to as *diagonal* those networks having diagonal microchannels in addition to the orthogonal ones (right Figure 2.1).

The optimization process starts from an initial structure that we call the *network template*, which is the equivalent to the ground structure used in topology optimization [15]. This network template has the maximum number of edges for the type of networks being designed, and is illustrated in Figure 2.2 for orthogonal and diagonal networks over a

Figure 2.2 – 2D template structures for orthogonal (left) and diagonal (right) networks.

6×6 point lattice. The optimization then consists in finding the diameter for each edge of the template structure from a finite set of discrete values $\mathcal{D} = \{D_0, D_1, \ldots, D_m\}$ such that the resulting networks minimize two or more objective functions subjected to a set of constraints. In contrast to the preliminary study on orthogonal networks presented in

[31], the present study also allows for diagonal microchannels. However, microchannels are permitted to intersect only on locations determined by the spatial point lattice in the optimization process, so that a network containing microchannels intersecting outside the point lattice (large hollow circle in Figure 2.1) is marked as unfeasible.

The solution of the flow objective functions requires boundary conditions. Figure 2.1 shows the inflow $\mathcal{S}$ and outflow $\mathcal{T}$ vertices located at the lower-right and upper-left corners of the domain, respectively. These are the boundary conditions considered in the 2D problems investigated in this work, although other configurations including multiple inflows and outflows can also be considered. In the more general case, inflow and outflow vertices are defined over sets $\mathcal{S} \subset \mathcal{V}$ and $\mathcal{T} \subset \mathcal{V}$, respectively. The formulation used to obtain the objective functions and constraints is described next.

2.2.1 Void volume fraction and intersection constraint

The presence of a microvascular network in a material can have a detrimental impact on its response to external loads as it tends to decrease its stiffness and strength. Thus, one of the requirements for these materials is to minimize the void volume associated with the microvascular network. For a 2D domain, the void volume fraction is defined as

$$v := \frac{\sum_{e \in \mathcal{E}} L_e D_e}{A_\Omega}, \tag{2.1}$$

corresponding to the normalized projected area of the network over its plane. Note that this definition does not correct for the overlapping areas between adjacent microchannels, which is assumed to be negligible for microchannels with high aspect ratios. Similarly, for 3D domains and cylindrical microchannels, the void volume fraction is defined as

$$v := \pi \frac{\sum_{e \in \mathcal{E}} L_e D_e^2}{4 V_\Omega} \tag{2.2}$$

When dealing with the optimization of diagonal network templates, there exists the possibility of producing networks where microchannels intersect at locations other than those determined by the point lattice. As mentioned earlier, these networks are considered to be unfeasible. Furthermore, the flow pressure values described in the next section are computed only at lattice point locations so the pressure drop resulting from those unfeasible networks is incorrect. Even though this problem could be avoided by creating a vertex at the microchannel intersection, it may be desirable from the manufacturing point of view to have the microchannels intersect at points determined by the point lattice. We

thus introduce the microchannel intersection constraint, ψ_I, as

$$\psi_I := \sum_{e_i \in \mathcal{E}} \sum_{e_j \in \mathcal{E}, e_j \neq e_i} \eta_{ij}^I, \tag{2.3}$$

where $\eta_{ij}^I = 1$ if microchannels e_i and e_j intersect at a location not determined by the lattice and $\eta_{ij}^I = 0$ otherwise.

2.2.2 Flow objectives and constraints

The flow efficiency is obtained by computing the maximum pressure drop between any two vertices in the spatial domain, assuming that the pressure drop in any microchannel e is given by the classical Hagen-Poiseuille law

$$\Delta p_e = \frac{128 \nu L_e}{\pi D_e^4} \dot{m}_e, \tag{2.4}$$

where ν and $\dot{m}_e$ denote the kinematic viscosity and mass flow rate of the fluid, respectively. Assembling the contribution of all microchannels in the network results in a linear system of equations $\vec{K}\vec{p} = \vec{\dot{m}}$, where $\vec{K}$ is the network characteristic matrix, $\vec{p}$ is the network pressure vector and $\vec{\dot{m}}$ is the network mass flow rate vector [32]. Boundary conditions for this problem involve prescribed pressure values at outflow locations and prescribed flow at the inflow. The flow efficiency of the network is thus given by

$$\Delta p := \max_{v_i, v_j \in \mathcal{V}} |p_{v_i} - p_{v_j}|. \tag{2.5}$$

Note that the pressure drop in a microchannel (Equation (2.4)) is inversely proportional to the fourth power of its diameter, in contrast to the linear or quadratic relationship found earlier for the void volume fraction (Equations (2.1) and (2.2)).

A good microvascular network design not only minimizes the energy needed to drive the flow through the network but also ensures that each of the microchannels has an active flow. This is particularly important for self-cooling or self-sensing applications. Thus, inner loops with no pressure differential have to be eliminated from the optimization, but the GA cannot infer this information from the pressure drop computation given earlier. As a result, we label as unfeasible those networks where there is zero flow in at least one

of their microchannels. To that purpose, we define the non-zero flow constraint, ψ_F, as

$$\psi_F := \sum_{e_i \in \mathcal{E}} \eta_i^F, \qquad \eta_i^F = \begin{cases} 1 & \text{if } \dot{m}_{e_i} = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

When dealing with multiple targets, there is a need to distribute the outflow evenly among the various targets. Consider that the resulting design of a square microvascular network is just a cell of a larger structure. Certainly, outflows that share a boundary of the network will become inflows for a contiguous cell, so having heterogeneous flows will introduce another complexity in the design of contiguous cells. On the other hand, if the healing process is started at outflow locations (as in the healing of a coating [8]), flow homogenization is also needed to obtain homogeneous healing. The fluid healing agent undergoes a chemical reaction with the catalyst that is present in the coating, but this reaction could be avoided completely if the flow were too high (washed out) or it could even block the outflow if the flow were too low. In all problems studied in this work, we assume that there is an external source of energy to drive the flow over the network, so the amount of fluid that leaves the system is proportional to the prescribed inflow. To ensure that the healing process at the $\mathcal{T}$ outflow locations be as homogeneous as possible, we define the flow homogeneity parameter, ζ, as

$$\zeta := \frac{1}{\dot{m}_{\mathcal{S}}} \max_{v_i, v_j \in \mathcal{T}} |\dot{m}_{v_i} - \dot{m}_{v_j}|, \tag{2.7}$$

where the maximum outflow difference is normalized by the total network inflow $\dot{m}_{\mathcal{S}}$.

2.2.3 Coverage and redundancy

In the case of self-healing polymers, an ideal microvascular network will cover the entire domain so that internal cracks nucleating anywhere in the domain can be arrested before reaching a critical length leading to a catastrophic failure. The coverage is hereafter imposed as a constraint by simply requiring a minimum degree $\bar{d}$ on each vertex of the graph that represents the microvascular network. The degree constraint, ψ_D, is then computed as

$$\psi_D := \sum_{v_i \in \mathcal{V}} \eta_i^D, \qquad \eta_i^D = \begin{cases} 1 & \text{if } \deg(v_i) < \bar{d}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

Probably the most desirable feature in the design of a microvascular network for self-healing applications is the reliability of its structure when subjected to damage. In other

words, when one or more microchannels are cut within the domain, the healing agent must be able to reach most if not all areas in a good network design. The severity of such a cut depends on the location of the damaged microchannels, ranging from preventing flow to a single vertex in a corner to completely preventing any vertex from receiving flow (in case the microchannel connecting the inflow is cut). This problem has been studied thoroughly in the literature and is designated as the *survivable network design problem* [33, 34]. The reliability of a network is obtained numerically through the redundancy matrix $\mathbf{r}$, where each entry r_{ij} represents the number of *node disjoint paths*[1] between vertices v_i and v_j. In other words, r_{ij} denotes the number of microchannels that have to be damaged in order to make vertex v_j unreachable from vertex v_i. It is well known that the redundancy matrix can be obtained by solving the maximum flow problem after assigning a unit capacity to all edges [33]. However, obtaining the redundancy matrix is computationally expensive since each entry r_{ij} has a complexity $O(\mathcal{V}\,\mathcal{E}^2)$ with the algorithm devised by Edmonds and Karp [35]. Since the computation of the minimum degree $\bar{d}$ is only of complexity $O(\mathcal{V})$, we have opted to investigate the relation between this parameter and the resulting network redundancy[2]. We use the observation that there must be at least $\bar{d}$ incident microchannels on vertices v_i and v_j if there are to be $\bar{d}$ disjoint paths between them. Nevertheless, this is just a necessary condition since r_{ij} might be lower than $\bar{d}$ even if both vertices have $\bar{d}$ incident microchannels.

To support this statement, let us consider the two groups of 2D networks considered in this study over a square domain, i.e., the orthogonal and diagonal microvascular networks. Furthermore, let us use the diameter set $\mathcal{D} = \{0, 1\}$ to indicate the presence or absence of a microchannel in the network. Both the dimensions of the domain and the magnitude of the diameters used are irrelevant to this study, which aims at correlating the redundancy of the network to its minimum degree. In any cartesian direction, there are $n = a/\delta + 1$ vertex locations. For a particular value of n, there are a total of $|\mathcal{D}|^{\mathcal{E}}$ possible microvascular networks, with $|\mathcal{D}| = 2$, $\mathcal{E} = n\,(n-1)$ for orthogonal networks and $\mathcal{E} = 2\,(2n-1)\,(n-1)$ for diagonal networks. The lattices considered range from 2×2 up to 6×6. It is clear that the exponential dependence on the number of edges, which in turn depends on n, makes the study of larger lattices prohibitively expensive. Note on the other hand that many of the networks within that total are not considered because they are either disconnected networks or, in the case of diagonal networks, there are microchannels that intersect in points not defined by the 2D lattice. Since the redundancy matrix is symmetric (i.e.,

[1] Two paths are called disjoint if they share no nodes other than the endpoints [34].

[2] The definition of $\bar{d}$ is in contrast to the formal definition of the degree of a graph, which is the maximum degree of its vertices.

$r_{ij} = r_{ji}$), let us define the network average redundancy as

$$\bar{r} := \frac{1}{\mathcal{V}(\mathcal{V}-1)} \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}, v_j \neq v_i} r_{ij}. \tag{2.9}$$

Figure 2.3 shows the average redundancy values for the feasible orthogonal (left) and diagonal (right) networks, where each data set includes only structures for a particular lattice size. For each set, the average of $\bar{r}$ over all networks in the survey is marked by a symbol, along with error bars that indicate maximum and minimum values. Results are slightly offset along the $\bar{d}$ axis to ease the visualization. For orthogonal networks (left Figure 2.3), a 5×5 lattice requires more than a trillion evaluations. As a result, a sample of the total number of networks was taken for the evaluations (indicated in the figures by an asterisk after the lattice size). The sampling size, ranging from 10^6 to 10^7 networks, was increased until no significant variation was obtained in the results. For diagonal networks (right Figure 2.3), a sampling was required even for a 4×4 lattice since a complete survey of all networks involves more than 4 trillion evaluations. The figures show a quasi linear dependence between the minimum degree of a network and its average reliability. In other words, a microvascular network with minimum degree $\bar{d}$ has a high probability of having $\bar{r} = \bar{d}$. This allows us to use the degree as a fundamental parameter in the design of microvascular networks.

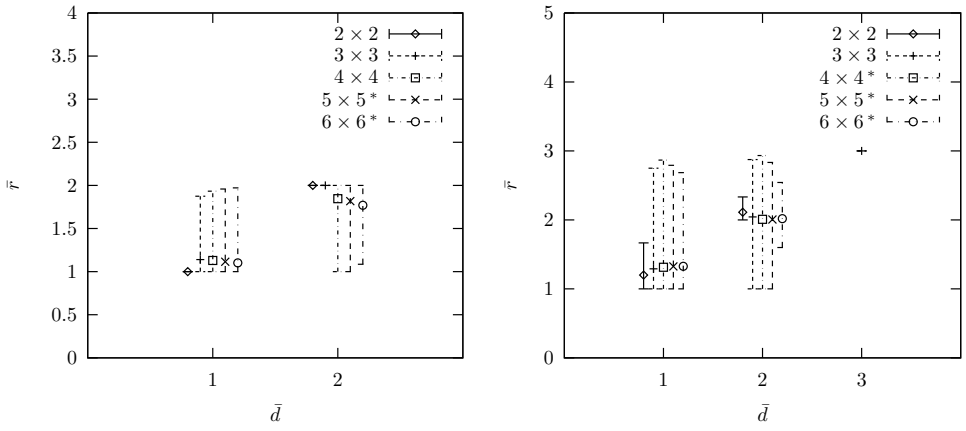


Figure 2.3 – Correlation between the minimum degree $\bar{d}$ and average network redundancy $\bar{r}$ (Equation (2.9)) for orthogonal (left) and diagonal (right) networks of various lattice sizes. For each data set, the average value of $\bar{r}$ for the survey is marked along with error bars indicating minimum and maximum values. An asterisk indicates that the data was obtained through sampling.

2.3 Genetic algorithms

Since their introduction more than four decades ago [36, 37], GAs have proven to be an efficient heuristics tool for the optimization of complex design problems. The notation adopted in this section follows the convention used in [38, 39]. GAs usually start the optimization from a population $\mathcal{P}$ containing a fixed number $|\mathcal{P}|$ of individuals $\mathcal{I}$, each with a distinctive chromosome $\mathcal{C}$. In order to use the algorithm, a suitable representation must be chosen to encode in the chromosome of an individual the different characteristics of the problem for which an optimum is sought. The chromosome is a numeric string where each location (allele $\mathcal{A}$ in GA terminology) encodes a parameter (or part of it) of the problem. The alleles can be chosen from a binary alphabet ($\mathcal{A}_i \in \{0, 1\}$), a $k-$ary alphabet ($\mathcal{A}_i \in \{0, 1, \ldots, k-1\}$), or from the real numbers ($\mathcal{A}_i \in \mathbb{R}$).

In a simple GA [16], the population of individuals $\mathcal{P}_0$ is initialized randomly, after which the algorithm enters a loop over $t_{\max}$ iterations or generations. In this loop, the entire population is first evaluated according to a prescribed objective function that describes numerically how fit the individuals are to solve the problem at hand. After this step another loop is initiated to determine the parent population for the next generation by applying the genetic operators *selection*, *crossover* and *mutation*. The selection mechanism is where the "survival of the fittest" takes place since those individuals that are more fit are more likely to be selected for reproduction. The mixing of the genetic information takes place in the crossover operator, where two selected individuals interchange the alleles of their chromosomes according to a certain criterion. The mutation is a slight modification in the characteristics of an individual and it is usually performed by changing the value of one of the alleles in the chromosome.

Even though the simple GA can be used to solve Multi-Objective Optimization Problems (MOOPs) [40], better results can be achieved by using Multi-Objective Evolutionary Algorithms (MOEAs) [41, 42, 43, 44]. With the simple GA, a MOOP is converted to a single weighted objective function, with different weights selected to assign different priorities to the corresponding objectives. The drawbacks of using a simple GA to solve MOOPs are evident not only because the solution obtained cannot capture the tradeoff between the different objectives but also because the determination of the weighting factors tends to influence the solution substantially. MOEAs, on the other hand, can capture the tradeoffs between conflicting objectives by carrying out a Pareto-optimization, in which individuals are compared in all objectives to determine the best candidates for the next generation. Some definitions are needed to understand how the MOEA-based optimization process works. At a given generation $t < t_{\max}$, let us divide the population in two mutually

exclusive sets containing feasible and unfeasible individuals, i.e.,

$$\mathcal{P} = \mathcal{P}^f \cup \mathcal{P}^u, \quad \mathcal{P}^f \cap \mathcal{P}^u = \emptyset,$$

where $\mathcal{P}^f = \left\{\mathcal{I}_i \mid \boldsymbol{\psi}_i = \mathbf{0}\right\}$ and $\mathcal{P}^u = \left\{\mathcal{I}_i \mid \boldsymbol{\psi}_i \neq \mathbf{0}\right\}$. Let the domination operator, $\ll_{\mathcal{S}}$, over the space $\mathcal{S}$ be defined such that the $i-$th individual $\mathcal{I}_i$ in the population $\mathcal{P}$ dominates another individual $\mathcal{I}_j$ if $\mathcal{I}_i$ is better than $\mathcal{I}_j$ in at least one dimension over the space $\mathcal{S}$ and is no worse in the rest[3]. Therefore, we denote $\mathcal{I}_i \ll_\Psi \mathcal{I}_j$ an individual $\mathcal{I}_i$ that dominates another individual $\mathcal{I}_j$ in the constraint function space Ψ. The same reasoning follows for the objective function space Φ. With this definition, let the global domination operator, $\lll$, be defined such that $\mathcal{I}_i$ dominates $\mathcal{I}_j$ (i.e., $\mathcal{I}_i \lll \mathcal{I}_j$) if any of the following conditions holds:

- $\mathcal{I}_i \in \mathcal{P}^f, \mathcal{I}_j \in \mathcal{P}^u$
- $\mathcal{I}_i, \mathcal{I}_j \in \mathcal{P}^u, \mathcal{I}_i \ll_\Psi \mathcal{I}_j$
- $\mathcal{I}_i, \mathcal{I}_j \in \mathcal{P}^f, \mathcal{I}_i \ll_\Phi \mathcal{I}_j$

In other words, the $i-$th individual dominates the $j-$th individual if $\mathcal{I}_i$ is feasible and $\mathcal{I}_j$ is not, or both individuals are unfeasible but $\mathcal{I}_i$ dominates $\mathcal{I}_j$ in constraint function space Ψ, or both individuals are feasible and $\mathcal{I}_i$ dominates $\mathcal{I}_j$ in objective function space Φ. With this definition, the evolution is different from a single-objective optimization since the selection process keeps nondominated solutions. This is schematically shown in Figure 2.4, where a randomly created population is optimized[4] for two objective functions ϕ_1 and ϕ_2. Through the evolution process, all individuals approach the so-called Pareto-optimal front, where all individuals lying on it are optimal solutions to the MOOP.

For this work, the Nondominated Sorting Genetic Algorithm II (NSGA-II) introduced by Deb et al. [46] is used. The original algorithm was modified in order to handle constraints by using the operator $\lll$ defined earlier, and is described in Algorithm 1. The main difference between the simple GA and the NSGA-II presented in Algorithm 1 is in the way individuals are selected in order to retain those nondominated solutions. An offspring population is obtained by applying the same genetic operators used for the simple GA, forming a combined parent-offspring population of size $2|\mathcal{P}|$. All individuals in the combined population are grouped first according to their non-domination rank in fronts F_i, $i = 1, 2, \ldots, n$, such that individuals in F_i dominate those of subsequent fronts (i.e., fronts $F_j, j > i$). In order to select the individuals that will constitute the parent

[3]The concept of dominance is borrowed from the field of economics and it is due to Vilfredo Pareto [45].

[4]Minimization is carried out in the schematic figure, thus the Pareto-optimal front is closer to the origin at the lower-left corner.

Algorithm 1 NSGA-II [46]

procedure EVOLVE($|\mathcal{P}|, t_{\max}$)
  $\mathcal{P}_0$ = RANDOM($|\mathcal{P}|$))
  Q_0 = APPLYGENETICOPERATORS($\mathcal{P}_0$)
  while $t < t_{\max}$ **do**
    $t \leftarrow t + 1$
    $R_t = \mathcal{P}_t \cup Q_t$
    F = FASTNONDOMINATEDSORT(R_t)
    repeat
      CROWDINGDISTANCEASSIGNMENT(F_i)
      $\mathcal{P}_{t+1} = \mathcal{P}_{t+1} \cup F_i$
    until $|\mathcal{P}_{t+1}| < |\mathcal{P}|$
    sort($\mathcal{P}_{t+1}, \lll$)
    $\mathcal{P}_{t+1} = \mathcal{P}_{t+1}[0 : |\mathcal{P}|]$
    Q_{t+1} = APPLYGENETICOPERATORS($\mathcal{P}_{t+1}$)
  end while
end procedure

population in the next generation, the algorithm first selects all fronts that completely fit in the population size (i.e., all fronts F_i are selected such that $|\cup_i F_i| \leq |\mathcal{P}|$). The $|\mathcal{P}| - |\cup_i F_i|$ remaining individuals (if any) are selected by also considering a crowding parameter, used to distribute individuals more evenly along the fronts throughout the computation. To illustrate how the algorithm works, consider the two-objective optimization problem MPO4 presented in [46]. The left plot in Figure 2.5 shows the 100 selected individuals, most of them lying in the Pareto-optimal front for this problem, while the right plot presents the resulting fronts obtained in that particular generation for the combined population of size $2|\mathcal{P}|$. Note that, on the figure that shows the fronts F_i, the discrete points are joined by lines for visualization purposes, but the lines themselves do not represent the fronts.

2.3.1 Microvascular network representation

As mentioned earlier, the use of GAs to carry out the optimization of a problem involves finding a suitable GA representation. In our case, the microvascular network is idealized as a graph data structure and the optimization is carried out on the diameters. An integer representation of cardinality k, referred to as k-ary alphabet in GA terminology, is chosen to represent the set of k possible discrete diameters. Therefore, the number of alleles in the chromosome of a *network individual* equals the number of microchannels in the network and the allele values map to a discrete set of diameters, chosen according to manufacturing

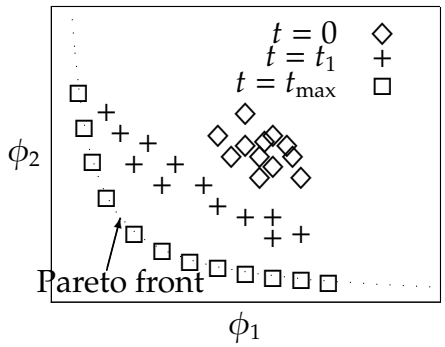


Figure 2.4 – Schematic of multi-objective optimization, showing a randomly created population of individuals at $t = 0$ evolving towards a Pareto-optimal front at iteration $t = t_{\max}$.

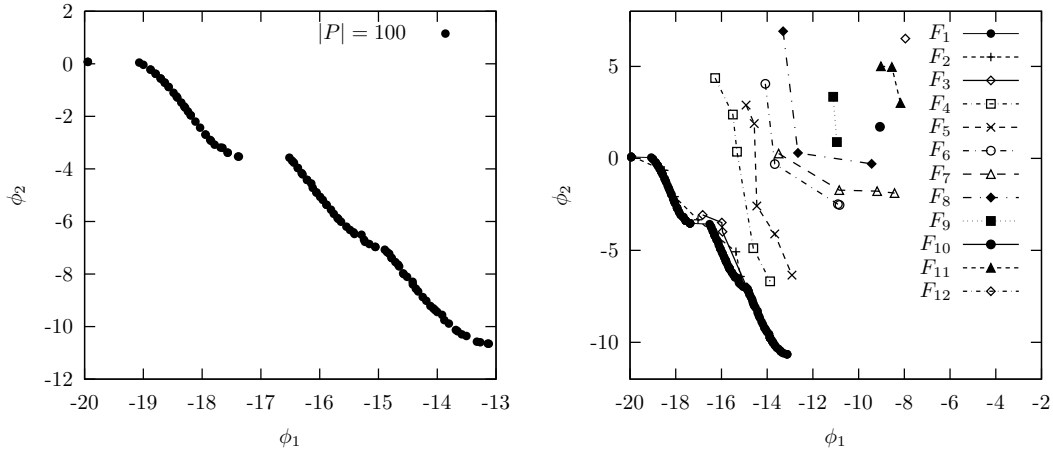


Figure 2.5 – NSGA-II results for problem MPO4 [46] at a particular generation, showing most individuals lying on the Pareto-optimal front for this problem (left) and the corresponding non-dominated fronts $F_i, i = 1, \ldots, 12$ (right). Discrete points representing the fronts in the right figure are joined by lines for visualization.

requirements.

To illustrate the network representation, let us consider a 2D diagonal network template on a 3×3 point lattice (left Figure 2.6). This template has 20 microchannels and a conforming chromosome will have the same number of alleles. Let the possible diameter choices be limited to the set $\mathcal{D} = \{D_i \mid D_i = i \cdot 25\,\mu\text{m}, i = 0, 1, \ldots, 5\}$. Then, the chromosome

$$C = \{44031205311400225042\}$$

produces the mapping shown at the right in Figure 2.6, i.e., the first and second alleles map to a diameter $D_4 = 100\,\mu$m, the third one maps to $D_0 = 0\,\mu$m (absence of the microchannel), the fourth one to $D_3 = 75\,\mu$m, and so forth.

Figure 2.6 – Network template (left) for diagonal networks over a 3×3 point lattice and resulting microvascular network (right) after applying the mapping given by a chromosome C.

2.3.2 Objective functions and constraints

The selection process within the GA needs to differentiate the "fittest individuals" by assigning numerical values to each of the objective functions and constraints considered. Most of the study presented in this manuscript focuses on the objective functions associated with the flow efficiency and void volume fraction, with an example considering also the flow homogeneity across outflow locations. The coverage of the network is imposed by constraint, as well as the requirement to have intersection of microchannels only at lattice point locations. Most examples will also enforce the presence of flow in every microchannel. In all cases presented in this paper, the optimization involves the minimization of the objective functions and is stated formally as

$$\begin{aligned} &\text{minimize} && \boldsymbol{\phi} = \left\{\phi_i\right\}_{i=1}^{m}, \\ &\text{such that} && \boldsymbol{\psi} = \{\psi_i\}_{i=1}^{n} = \mathbf{0}, \end{aligned} \tag{2.10}$$

where the optimization is carried out on the set of m objective functions subjected to n equality constraints.

The void volume fraction is computed using Equations (2.1) and (2.2) for the 2D and 3D networks, respectively. The flow efficiency is evaluated through the computation of the maximum pressure difference between any two vertices of the network, i.e., Equation 2.5. The computed values are normalized by a reference network topology. For 2D problems involving a square domain of side a with a single inflow and outflow located at opposite corners (right Figure 2.1), the reference network consists of a single microchannel connecting the source to the target with diameter $D_{\text{ref}} = 100\,\mu\text{m}$. The corresponding reference values for the void volume fraction (v_{ref}) and pressure drop (Δp_{ref}) are

$$v_{\text{ref}} = \frac{\sqrt{2} D_{\text{ref}}}{a}, \qquad \Delta p_{\text{ref}} = \frac{128\sqrt{2} a \nu}{\pi D_{\text{ref}}^4} \dot{m}_S. \tag{2.11}$$

For more complex 3D problems with single or multiple outflow locations, the reference network topology is chosen to be the same as the template used in the optimization.

Thus, the objective functions associated with the void volume fraction, ϕ_v, and flow efficiency, ϕ_f, are

$$\phi_v = \frac{v}{v_{\text{ref}}}, \qquad \phi_f = \frac{\Delta p}{\Delta p_{\text{ref}}}, \tag{2.12}$$

where v and Δp correspond to the values of the network being evaluated. A third objective function considered in this study is the flow homogeneity defined by Equation (2.7) (i.e., $\phi_h = \zeta$), where the maximum flow difference among outflows is normalized by the total inflow. Since zero diameter (i.e., missing) channels are considered in the optimization process, disconnected networks may occur. Where such a network is found, very high values are assigned to ϕ_f and ϕ_h so the network becomes a "bad individual" for the selection process in the GA.

2.3.3 Genetic operators

The creation of an offspring population in the NSGA-II algorithm to form the combined population involves the use of genetic operators. For the selection process, tournament selection with selection pressure $s = 4$ is used [17]. In tournament selection, two individuals are selected and the dominated individual according to the operator $\lll$ defined in Section 2.3 is discarded. The winner is then compared to another contender and this process is repeated until s individuals are compared.

One point crossover is used in all crossover operations [16]. This mechanism randomly

selects one of the $|C| - 1$ possible crossover points and then exchange all alleles between the two selected individuals after this point. A high probability of crossover p_c is used in this work, usually $p_c = 0.9$.

The chromosomes in this study have cardinality $k > 2$ so simple bit mutation, the most common mutation operator [16], cannot be used. The mutation operator considered here uses a Gaussian probability density function centered at the value of the allele to be mutated. In this way, the mutation has a higher probability to select values that are close to the original allele. This technique is borrowed from other evolutionary algorithms and is adapted so it can be used with integer values. The probability of mutation p_m is chosen such that, on average, at most one allele is mutated in the chromosome, i.e., $p_m \leq 1/|C|$.

2.4 Results

This section starts with the description of a typical multi-objective optimization using GAs, followed by a series of optimized problems. If not explicitly stated, it is implied that the GA has completely converged to a Pareto-optimal front. This can be concluded after observing no significant differences on the front for many generations.

2.4.1 Multi-objective constrained optimization

To illustrate the optimization process, consider a 2D square domain with side $a = 10\,\text{mm}$ where the point lattice has $\delta = 1\,\text{mm}$ (11×11 lattice), resulting in $\mathcal{V} = n^2 = 121$ vertex locations, with $n = a/\delta + 1$. The set of possible diameter choices is $\mathcal{D}\,[\mu\text{m}] = \{0, 100, 200\}$. The inflow and outflow vertices are located in the lower-right and upper-left corners, respectively. A diagonal template is considered for the optimization, containing $\mathcal{E} = 2\,(2n-1)\,(n-1) = 420$ edges. We constrain the optimization to those networks having microchannel intersections at lattice point locations (Equation (2.3)), covering the entire domain with a minimum degree $\bar{d} = 2$ (Equation (2.8)) and with flow at every microchannel (Equation (2.6)). Considering as objectives for the optimization the void volume fraction and the flow efficiency, the resulting optimization problem is stated as

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{\phi} = \left\{\phi_v, \phi_f\right\}, \\ \text{such that} \quad & \boldsymbol{\psi} = \{\psi_D, \psi_I, \psi_F\} = \mathbf{0}. \end{aligned}$$

A typical run of the multi-objective optimization is illustrated in Figures 2.7, which

present snapshots of the objective function and constraint spaces at generations $t = 0$, $t = 21$ and $t = 25$. The initial randomly generated population contains only unfeasible individuals, as shown in Figures 2.7a and 2.7d. Both plots show only those individuals for which the microvascular network is not disconnected, accounting for approximately 71% of the total. Recall that when a disconnected network is found, the flow efficiency objective and the non-zero flow constraint are penalized with a very high value (i.e., machine representation of infinity). At this stage the optimization is carried out in constraint space and the algorithm selects the better structures for the next generation according to their feasibility. Generation $t = 21$ is the first generation for which a feasible individual appears (indicated by an arrow in Figure 2.7b) and the entire population becomes feasible after only four more generations as shown in Figures 2.7c and 2.7f. After this point, the optimization is carried out in objective function space until the algorithm converges to the Pareto-optimal front. It is worth mentioning that unfeasible individuals are completely eliminated from the population after a few generations for all objective functions and constraints presented in this work. Moreover, no new unfeasible solutions appear after the entire population becomes feasible.

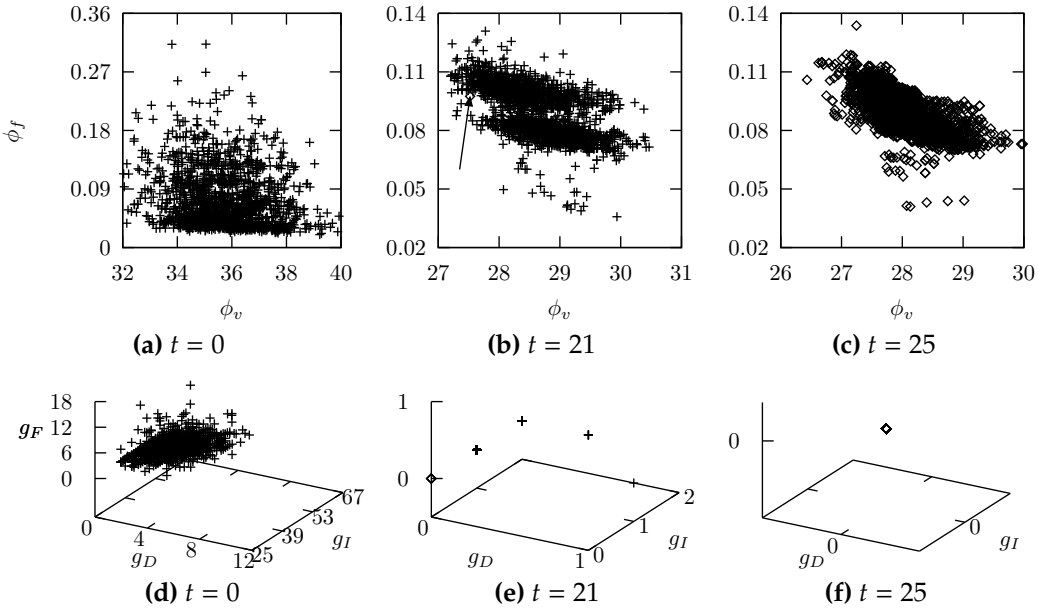


Figure 2.7 – Objective function and constraint spaces at three early steps of the optimization ($t = 0$, $t = 21$ and $t = 25$). Plots on the upper row show the objective function space composed of void volume fraction ϕ_v and flow efficiency ϕ_f. Plots on the lower row show the constraint function space, composed of degree constraint ψ_D, intersection constraint ψ_I and flow constraint ψ_F. Crosses (+) and diamond ($\diamond$) symbols indicate unfeasible and feasible individuals, respectively.

Optimization results after 2000 generations for a population size $|\mathcal{P}| = 5000$ are presented in Figure 2.8 (top). The obtained Pareto-optimal front for competing objectives ϕ_v and ϕ_f is shown on the left, whereas the microvascular networks for selected individuals labeled (a) through (d) are shown on the right. Individual (a), which is composed only of 100 μm diameter microchannels, has the minimum void volume fraction but the highest pressure drop. Individual (b) shows a set of connected microchannels with the largest diameter $D = 200\,\mu$m directing most of the flow through a direct path between the source and the target. The emergence of this path in the solution creates a jump in the flow efficiency in the Pareto-optimal front. Similarly, the network corresponding to individual (c) has three main paths of microchannels with the largest diameter. Finally, individual (d) is the right-most individual resulting from this particular optimization, having the maximum void volume fraction, but the minimum pressure drop. It is important to note that all the networks obtained in this optimization are optimal solutions to the problem and that all of them respect the imposed constraints. Moreover, the algorithm picks the best orientation for most of the diagonal microchannels in the resulting optimized structures even though the starting template structure is not biased regarding the orientation.

Constraints reduce the search space and guide the GA to obtain only networks with desired properties and/or without undesired features. For example, consider the same optimization problem subjected this time to constraints $\boldsymbol{\psi} = \{\psi_D, \psi_I\}$, i.e., without the non-zero flow constraint. The results for this optimization are illustrated in Figure 2.8 (bottom), where the darker colored microchannels of the selected networks correspond to those that have no flow. In fact, for this particular run 90% of the resulting optimized structures had microchannels with zero flow. Interestingly, the Pareto-optimal front in this optimization is almost identical to that obtained while accounting for the non-zero flow constraint (top-left Figure 2.8), but the resulting structures have a completely different behavior.

2.4.2 Effect of microchannel orientation

The effect of the microchannel orientation is studied here by comparing the optimization of an orthogonal network template (left Figure 2.2) with the fully constrained results of the previous section. Thus, the optimization of the orthogonal template uses the same domain size and point lattice described earlier (i.e., $a = 10\,mm, \mathcal{V} = n^2 = 121$ vertex locations, $n = a/\delta + 1$). Void volume fraction and flow efficiency are used again as the objective functions. The orthogonal template has a total of $\mathcal{E} = 2n\,(n-1) = 220$ edges and is optimized using the diameter set $\mathcal{D}_O\,[\mu\text{m}] = \{100, 200\}$. There is no need to constrain

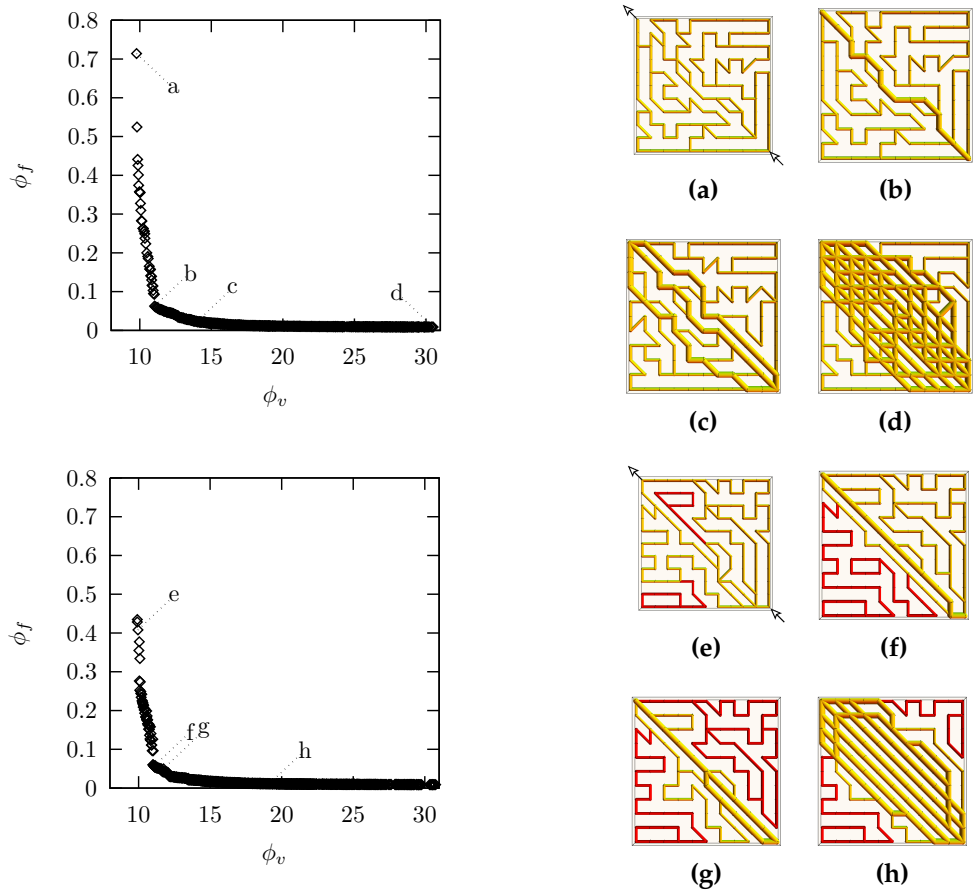


Figure 2.8 – Pareto-optimal front and selected networks after 2000 generations for a 11×11 square lattice with $\bar{d} = 2$, considering non-zero flow constraint (top), and neglecting it (bottom). Inflow and outflow vertices are located in the lower-right and upper-left corners, respectively. Eight networks labeled (a) through (h) on the optimal fronts are shown on the right. The darker segments in networks (e) through (h) denote microchannels with zero flow. The source and target locations are denoted by arrows in Figures 2.8a and 2.8e.

this optimization because the absence of the null diameter ensures complete coverage and intersections only occur at lattice points. Therefore, the optimization of the orthogonal template is stated as

$$\text{minimize} \qquad \boldsymbol{\phi} = \left\{\phi_v, \phi_f\right\}.$$

Recall from the previous section that the diagonal template considers also the null diameter, allowing for a possible match between the resulting diagonal and orthogonal networks in certain areas of the domain.

The NSGA-II results for both optimization problems are plotted in Figure 2.9, where they are denoted as *orthogonal* and *diagonal*. Different population sizes and maximum number of generations were chosen for each optimization. The chromosome size and its cardinality (number of possible allele choices) are higher for diagonal networks, so the genetic algorithm needs more computational resources to obtain solutions lying on the Pareto-optimal front. The optimization of orthogonal networks was run for 1000 generations with a population size $|\mathcal{P}| = 3000$. The orthogonal data set shows the entire population on the Pareto-optimal front. Recall from the previous section that the optimization of the diagonal network was run for 2000 generations with a population size $|\mathcal{P}| = 5000$. Not all the individuals of the diagonal data set are displayed as the figure is cropped for visualization purposes.

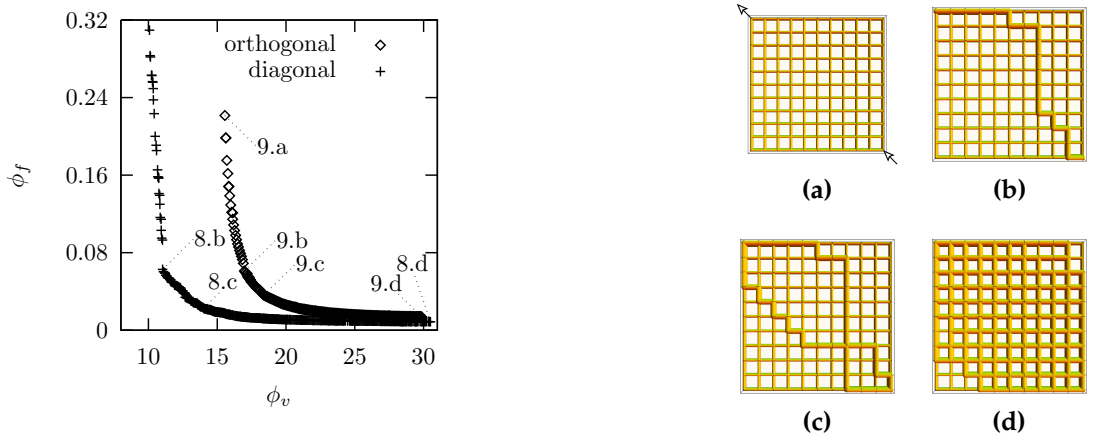


Figure 2.9 – NSGA-II results for the optimization of orthogonal and diagonal network templates (left) and selected microvascular networks for the orthogonal template optimization (right). Refer to Figures 2.8b through 2.8d for selected diagonal networks. Figure 2.9a shows the inflow and outflow locations.

Four networks from the orthogonal template optimization, labeled (a) through (d) on the Pareto-optimal front, are selected for visualization (right Figure 2.9). Labels 2.8b

through 2.8d refer to the diagonal networks presented in Figure 2.8 for the fully constrained optimization. As in the previous section, Figure 2.9a corresponds to the individual with the minimum void volume fraction but with the highest flow resistance. All the microchannels in the microvascular network converged to the minimum value $D = 100\,\mu$m. Solutions 2.9b and 2.9c resemble those given in the previous section, where most of the flow is directed through main paths of interconnected microchannels having the largest diameter. An individual for which almost all microchannels have converged to the largest diameter is presented in Figure 2.9d. The optimum individual with the highest void volume fraction (and consequently the minimum flow resistance) would have all its microchannels with the largest diameter, thus it is expected that running the algorithm for more generations or with a higher population size would produce this result. Note that having diagonal microchannels and the null diameter in the diagonal template optimization allows us to obtain individuals in a wider range for both objective functions and a much more efficient flow network for a given void volume fraction.

2.4.3 Effect of network minimum degree

We now turn our attention to the issue of coverage imposed as a constraint by requiring a minimum degree $\bar{d}$ on all vertices of the graph (Equation (2.8)). It was shown in Section 2.2.3 how this parameter relates to the reliability of the network when subjected to damage. We vary the minimum degree target $\bar{d}$ from 0 (unconstrained case) to 3 (networks having at least three incident microchannels at each vertex) for networks within a 2D square domain with side $a = 10\,$mm over a 6×6 point lattice ($\delta = 2\,$mm). Refer to Figure 2.2 for the template network used in the optimization and to Section 2.3.2 for a description on the reference network used for the normalization of the objective function values. For this study, the diameter set is $\mathcal{D}[\mu\text{m}] = \{0, 100\}$. Again, the inflow and outflow vertices are located at the lower-right and upper-left corners, respectively. The non-zero flow constraint is not imposed in all optimizations of this section.

Figure 2.10 presents the resulting Pareto-optimal fronts for all optimizations. For high values of the void volume fraction, all curves lie on top of each other since the distinction between different values of $\bar{d}$ vanish for "fully populated networks". For low values of ϕ_v, notable differences are obtained, with low-$\bar{d}$ networks achieving better flow efficiencies. Some of the resulting networks, labeled (a) through (l) on the fronts, are also shown in Figure 2.10. The first four figures correspond to the unconstrained case ($\bar{d} = 0$). The optimal solution for an individual containing the minimum void volume fraction would correspond to a set of microchannels forming a straight path between the source and the

target. The network shown in Figure 2.10a has a structure similar to that of the reference network used for normalization and the algorithm would have matched it had it been run for more generations or had the population size been increased. The individual shown in Figure 2.10d matches the optimum containing the highest void volume fraction. Note once again how the optimized networks naturally align with the source to target direction. Networks 2.10b and 2.10c correspond to two individuals with intermediate objective function values. Structures 2.10e and 2.10f belong to the second optimization, with imposed minimum degree $\bar{d} = 1$. The first network shows how the flow efficiency is deteriorated by maximizing the distance between inflow and outflow. The second network shows a jump in the pressure drop in Figure 2.10 due to the fact that this is the first network for which the flow reaches the outflow in two distinct paths. The networks resulting from the optimization with an imposed degree $\bar{d} = 2$ are labeled 2.10g through 2.10j. The first network shows the individual with minimum void volume fraction. The next three individuals show symmetric patterns that result from the optimization in increasing order of void volume fraction. The last two figures, labeled 2.10k and 2.10l, illustrate the results for the last optimization for which $\bar{d} = 3$. Note that the algorithm has oriented the two microchannels in the lower-left and upper-right corners of the lattice in order to respect the constraint on the degree. These two networks correspond to the extreme point locations on the Pareto-optimal front for this optimization.

2.4.4 Diameter set study

The influence of the magnitude of one of the diameters in the set as it increases from $D = 20\,\mu$m to $D = 80\,\mu$m is studied in this section for diagonal networks. Thus, the $i-$th optimization has a set of diameters

$$\mathcal{D}_i\,[\mu\text{m}] = \{0, 20\,(i-1)\,, 100, i = 1, 2, \ldots, 5\}\,.$$

As in the previous case, the square domain for this problem has a side $a = 10\,$mm, and spacing $\delta = 2\,$mm. Also, let us consider as feasible only those networks with imposed degree $\bar{d} = 2$, which guarantees a complete coverage of the domain and same level of redundancy.

Once again, the results obtained from the algorithm are shown in Figure 2.11, which are very similar to those given in the previous section since all curves lie on top of each other for high values of void volume fraction. The effect of the intermediate diameter value is seen for small values of ϕ_v. The figure was cropped for visualization purposes, as

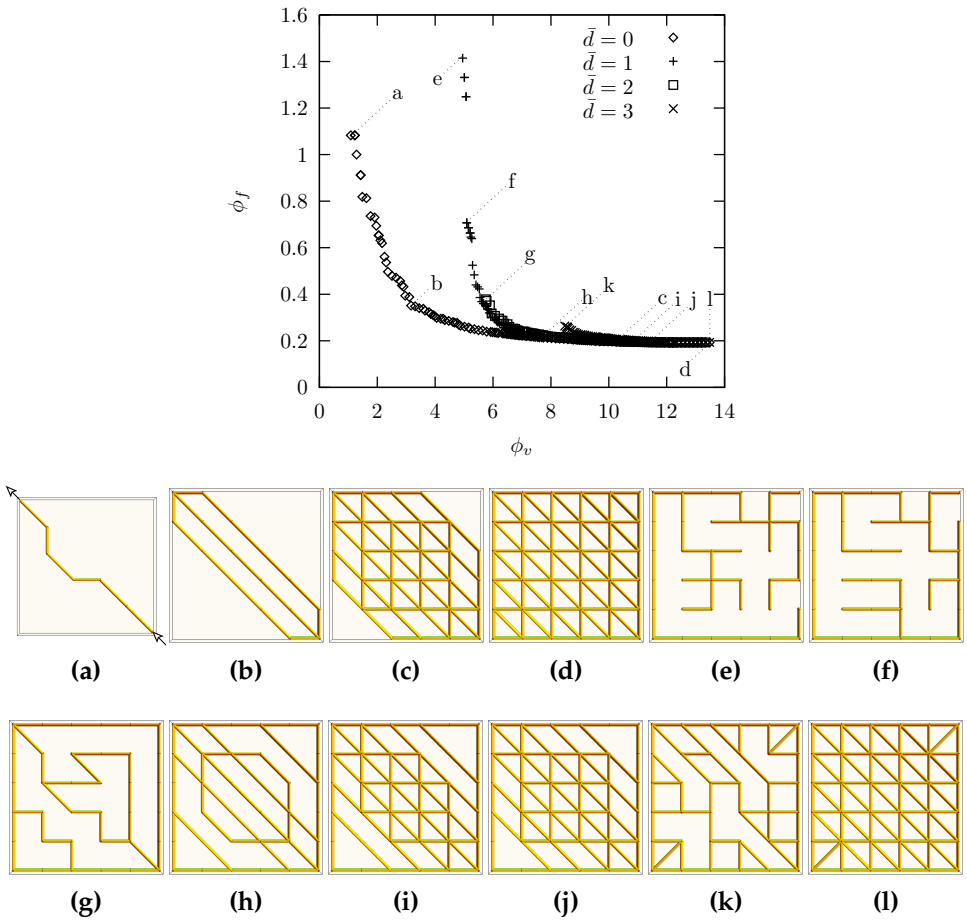


Figure 2.10 – NSGA-II results for network minimum degree study (top) and selected networks labeled (a) through (l) in the Pareto-optimal fronts. (a) through (d): $\bar{d} = 0$; (e) and (f): $\bar{d} = 1$; (g) through (j): $\bar{d} = 2$; (k) and (l): $\bar{d} = 3$. Source and target locations are marked by arrows in Figure 2.10a.

some networks composed almost exclusively of intermediate-diameter microchannels are characterized by very high pressure drop values. Recall that the pressure drop is inversely proportional to the fourth power of the diameter. The results presented in Figure 2.11 indicate that the most efficient networks combine large-diameter microchannels directing most of the flow between the source and the target and small-diameter ones addressing the coverage and redundancy requirement, as is the case in many biological systems.

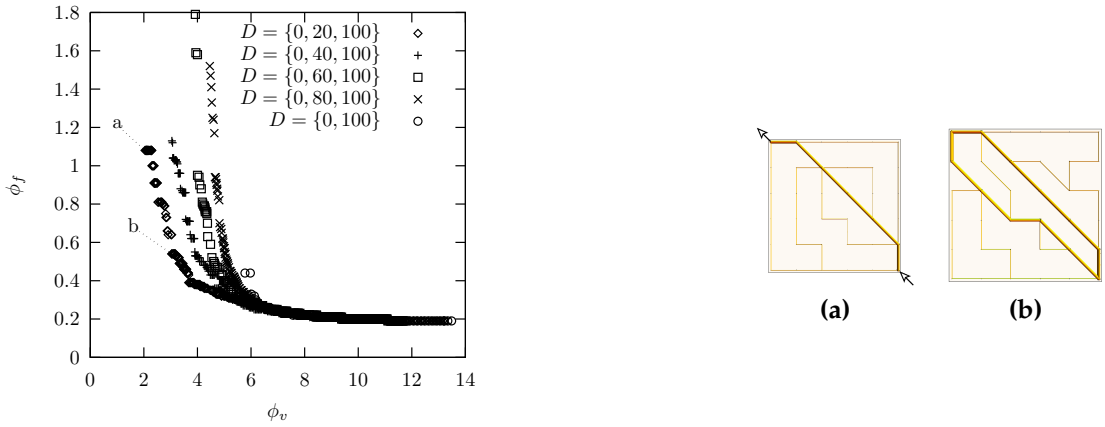


Figure 2.11 – NSGA-II evolution results for the diameter set study. Networks labeled (a) and (b) are shown on the right and correspond to the optimization with diameter set $\mathcal{D}[\mu\text{m}] = \{0, 20, 100\}$. The source and target locations are marked as arrows in Figure 2.11a.

2.4.5 3D optimization

A 3D domain is optimized in this section by simply changing the template structure. This can be done because the underlying optimization scheme using GAs is independent of the network used as the template. As usual, we take ϕ_v and ϕ_f as primary objective functions. Consider a cuboid with $a = b = c = 10\,\text{mm}$ (i.e., a cube) and a $11 \times 11 \times 11$ point lattice ($\delta = 1\,\text{mm}$). The template and reference networks have the same structure and consist of microchannels oriented only in the directions of the coordinate axes x, y and z (i.e., orthogonal network). For this structure, there are $\mathcal{E} = 3n^2(n-1) = 3630$ edges, where $n = a/\delta + 1$. The diameter set considered is again $\mathcal{D}[\mu\text{m}] = \{0, 100, 200\}$ with a fixed diameter for the reference structure $D_{\text{ref}} = 150\,\mu\text{m}$, resulting in $v_{\text{ref}} = 7.63\%$. For this problem the inflow is located at one of the vertices of the cube and the outflow is located at the vertex furthest from the inflow. The optimization is constrained to those networks having non-zero flow in all microchannels and having a minimum degree $\bar{d} = 2$.

Results for this optimization are summarized in Figure 2.12, where some structures

lying on the optimal front (top) are selected for visualization (bottom). Darker colored microchannels have the largest diameter value $D = 200\,\mu$m. The results follow the same criteria described in the previous sections, where the emergence of sets of microchannels having the largest diameter reduce the pressure drop but increase the void volume fraction. It is worth mentioning that this problem has a total of 3^{3630} different networks and yet the trade-off between conflicting objective functions is still well captured with a relatively modest population size of 8000 individuals.

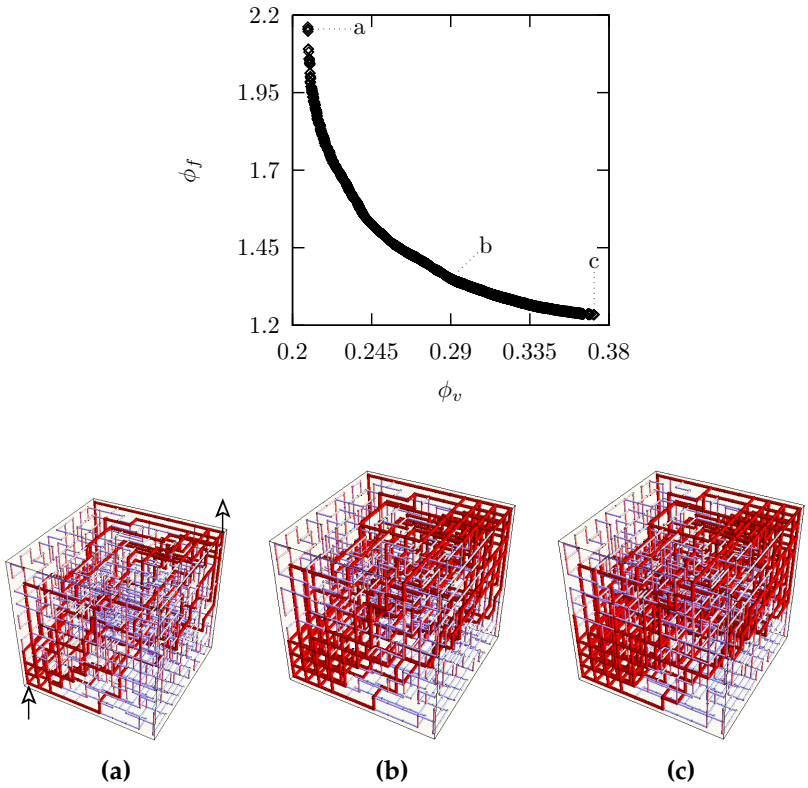


Figure 2.12 – NSGA-II results for a 3D optimization (top) and selected networks lying on the Pareto-optimal front (bottom). Darker-colored microchannels correspond to those with the largest diameter. The arrows in Figure 2.12a show source and target locations.

2.4.6 3-objective optimization

A third objective function that deals with the homogeneity of flow at target locations is introduced here. Consider a 3D cuboid with dimensions $a = 10\,$mm, $b = 4\,$mm and $c = 1\,$mm. A point lattice with $\delta = 1\,$mm produces $11 \times 5 \times 2$ possible vertex locations. Furthermore, consider all vertices lying on the top surface of the cuboid as outflows

(totalling $T = 55$ targets) and a single inflow is located at the midpoint of a short edge at the bottom surface. The template and reference network for this optimization are again orthogonal networks. The reference network has a diameter $D_{\text{ref}} = 50\,\mu$m and the diameter set to consider is $\mathcal{D}\,[\mu\text{m}] = \{0, 10, 100\}$. The minimization problem is then

$$\begin{aligned} &\text{minimize} && \boldsymbol{\phi} = \left\{\phi_v, \phi_f, \phi_h\right\}, \\ &\text{such that} && \boldsymbol{\psi} = \{\psi_D, \psi_F\} = \mathbf{0}. \end{aligned}$$

For this problem the degree constraint is relaxed at outflow locations because failing to do so would always produce unfeasible networks. In other words, all outflow vertices have the same pressure so any microchannel connecting two of them would have had zero flow had the degree constraint been satisfied.

Two different views of the resulting "Pareto-optimal surface" are illustrated in Figure 2.13, along with selected network structures. Once again the algorithm is able to capture the tradeoffs between the different objectives. After a few generations the algorithm eliminates all unfeasible solutions. This translates in assigning a zero diameter to all microchannels of the template close to the upper surface of the cuboid since all of them have zero flow. The network labeled 2.13a has the minimum value $\phi_h = 0.0005$ for the flow homogeneity objective function, showing that all vertical microchannels take the lowest non-zero diameter and a distribution network is created underneath. This network achieves a nearly optimum outflow distribution, with minimum and maximum outflow values deviating by only -1.25% and 1.46% respectively, from the optimum value $\dot{m}_S/T$. As shown in Figure 2.13 (top), the optimal front has a distinct discrete nature and is composed of two main "branches". The solution labeled 2.13b belongs to a set of solutions where ϕ_h is not optimal. However, these solutions are characterized by a substantially better flow efficiency as one or more large-diameter microchannels link the source to at least one of the targets. Finally, the network 2.13c corresponds to the individual with the lowest void volume fraction for this particular run, showing most of the microchannels converged to the smallest non-zero diameter. However, this network is not as optimal as network 2.13a in terms of outflow uniformity ($\phi_h = 0.1053$) and pressure drop ($\phi_f = 203.1$).

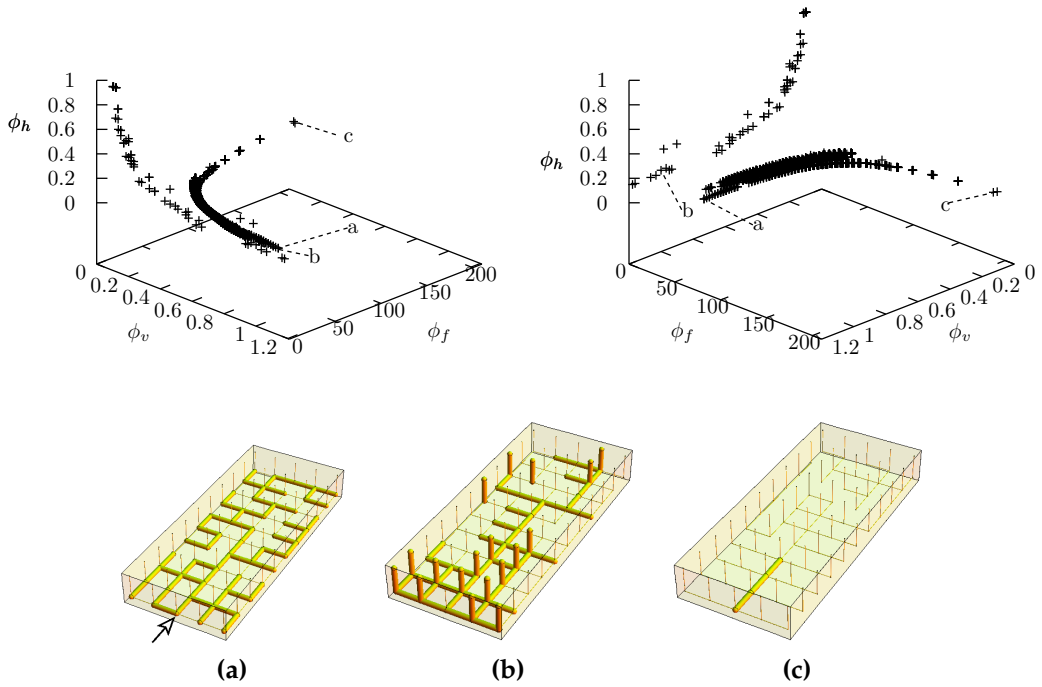


Figure 2.13 – NSGA-II results showing two different views of the Pareto-optimal surface for a 3-objective problem (top) and selected resulting structures (bottom). The boundary conditions for the flow objectives consider a single inflow shown by an arrow in Figure 2.13a and 55 outflows located in the upper surface.

2.5 Conclusions

This work has presented a study on the use of a multi-objective GA in the optimization of microvascular flow networks decomposed into a set of mathematically simple objective functions. The algorithm is able to capture adequately the tradeoff between conflicting objective functions and to eliminate unfeasible solutions. Constraints restrict the search space and guide the algorithm to obtain only feasible solutions after a few generations. The GA naturally evolves the networks towards paths of minimal flow resistance between the source and the target. The optimized results show how the addition of diagonal microchannels in the template improves flow efficiency, specially at low values of the void volume fraction. The analysis also demonstrated the cost in terms of void volume fraction of added redundancy in the networks. Similar to many biological systems, the most flow-efficient networks resulting from the GA optimization combine large-diameter microchannels directing most of the flow between the source and the target with small-diameter ones addressing the coverage and redundancy requirements. Even though the study was carried out on simple geometries, the extension to more complex geometries is easily accomplished by selecting a different template structure since the underlying evolution process is independent of the template selection, as demonstrated in the optimization of 3D templates.

To conclude this study, let us mention a couple of issues associated with the multi-objective GA used in this microvascular network design. In all problems studied in this work, the NSGA-II algorithm has no difficulty reaching the Pareto-optimal front and, once this front is reached, the subsequent generations create individuals that spread uniformly along it. However, the use of the crowding distance parameter in the selection exchanges optimal solutions in tightly crowded areas over the front for non-optimal yet less crowded ones [47]. Also, the algorithm has difficulties when trying to find the individuals with extreme values on the Pareto front. Furthermore, the algorithm does not scale well with the problem size. However, there have been successful attempts to obtain scalable GAs by combining the algorithm with other techniques [48] or by exploiting the "embarrassingly parallel" nature of the evaluation process [49].

3 GFEM-based solver for the heat equation

This chapter presents the peer-reviewed article published in 2009 in the *International Journal for Numerical Methods in Engineering* [19].

Generalized finite element enrichment functions for discontinuous gradient fields

Abstract

A general GFEM/XFEM formulation is presented to solve two-dimensional problems characterized by C^0 continuity with gradient jumps along discrete lines, such as those found in the thermal and structural analysis of heterogeneous materials or in line load problems in homogeneous media. The new enrichment functions presented in this paper allow solving problems with multiple intersecting discontinuity lines, such as those found at triple junctions in polycrystalline materials and in actively cooled microvascular materials with complex embedded networks. We show how the introduction of enrichment functions yields accurate finite element solutions with meshes that do not conform to the geometry of the discontinuity lines. The use of the proposed enrichments in both linear and quadratic approximations is investigated, as well as their combination with interface enrichment functions available in the literature. Through a detailed convergence study, we demonstrate that quadratic approximations do not require any correction to the method to recover optimal convergence rates and that they perform better than linear approximations for the same number of degrees of freedom in the solution of this type of problems. In the linear case, the effectiveness of correction functions proposed in the literature is also investigated.

3.1 Introduction

Discontinuous gradient fields appear in many problems in physics and engineering. Examples include thermal and structural analyses of heterogeneous materials such as polycrystals, where C^0 continuity is observed along grain boundaries, or composite materials,

where a discontinuous gradient is obtained along inclusion boundaries. Homogeneous materials can also exhibit solutions characterized by discontinuous gradients. Examples include problems with thermal or structural loads applied over very narrow regions, mathematically modeled as line loads. An example of such a model can be found in the thermal response of a new type of polymeric materials that contains an embedded microvascular network (i.e., a flow network where the diameter of the channels can be as small as 10 μm). These materials are currently being considered for thermal management [22]. In the mathematical model of such a system, the cooling effect of the microchannels can be collapsed to a thermal heat sink over a line.

The standard finite element method usually approaches these problems using a conforming finite element mesh [50]. Throughout the paper the terminology *conforming mesh* and *matching mesh* will be used interchangeably, referring to meshes where the edges of the finite elements follow the grain boundaries or the line loads[1]. The inherent C^0 continuous nature of the resulting finite element approximation automatically satisfies the required jumps in the gradient along those boundaries. However, there may be cases where creating a conforming mesh is not feasible or is computationally too demanding. The problem geometry can be such that the creation of a conforming mesh requires finite elements with unacceptable aspect ratios. Furthermore, creating a conforming mesh may necessitate advanced meshing tools not available to the analyst (especially in 3D) or sometimes not sufficiently robust to handle complex geometries. The complexity and computational cost of creating conforming meshes are especially critical in transient problems involving moving line loads or boundaries.

By eliminating the complexity of the computational geometry and allowing the discretization to become independent of the underlying geometry, the Generalized/Extended Finite Element Method (GFEM/XFEM) provides an attractive alternative for this class of problems. Since its introduction in the mid-nineties [51, 52, 53, 54, 55, 56], the method has increasingly gained attention in the FE community because of the added flexibility it offers compared to the conventional FEM. For more details on the history of development of these methods we refer the reader to [57] and the references therein. The GFEM/XFEM allows the use of *a priori* knowledge about the solution of a problem to obtain an improved finite element approximation or to recover optimal convergence by the use of a non-matching mesh. This knowledge is introduced through the use of *enrichment functions* that can range from polynomials to very sophisticated handbook functions (e.g., the Westergaard solutions for a crack in an infinite plate). The mesh independence that the method

[1] In contrast to other commonly used terminology, where a non-conforming mesh implies, e.g., regions in the mesh having hanging nodes and thus creating a discontinuous solution.

provides plays a fundamental role in problems that require complete remeshing or even refinement around areas of interest inside the problem domain. Problems addressed by the GFEM include structural problems [58, 59], crack propagation in fracture mechanics [60, 61, 62, 63, 64] and phase interface/change problems [65, 66, 67, 68].

The method has also been used for problems that involve embedded particles or holes [69, 70, 71]. In this line of work, the material interfaces completely cut the finite elements and the proposed enrichment functions attempt to recover the discontinuous gradient field at the interfaces. In order to do so, the elements split by the interfaces are subdivided in integration elements that use the right material properties according to the side of the interfaces they lie on. Material interfaces can also be handled using the GFEM proposed by Babuška and Osborn [72]. They used the so-called broken function to solve 1D problems using finite element meshes that do not match material interfaces. Most of the work available in the literature deals with these problems, and it was reported that some of these enrichments provide optimal convergence when the mismatch between material properties is not too high [73]. However, little attention has been paid to the more general case where multiple interfaces meet inside a finite element and a C^0 continuous field is recovered. In these cases, a conforming mesh to the junctions could be used in conjunction with the interface enrichment functions mentioned above. Yet, the use of a truly non-conforming mesh is always desired, and two approaches have been proposed in the literature to deal with multiple intersecting weak discontinuities. The first one uses enrichment functions based on the product of the distance functions to the interfaces [74]. The second approach, presented in [75], uses Heaviside enrichments and the continuity is enforced using a traction separation law, following the methodology proposed in [59] for strong discontinuities. This work introduces new enrichment functions that address the problem of having multiple interfaces intersecting inside finite elements and detailed convergence results are given. The end result is the creation of a finite element mesh that is completely independent of the geometry of the problem. The results in this paper are obtained in the context of the heat equation, but the enrichment functions are general and can be used to simulate other physical phenomena (e.g., elasticity problems). It is shown that in all cases, quadratic approximations are more accurate than their linear counterparts for the same number of degrees of freedom. It is also shown that the use of the correction to the GFEM/XFEM proposed in [76] is not required for quadratic approximations in order to achieve optimal convergence rates and that for linear approximations, some enrichment functions fail to recover optimal convergence even when using such correction. A detailed review on the GFEM/XFEM for material modeling can be found in [57].

Section 3.2 describes the problem to solve and gives a brief introduction to the GFEM.

The proposed enrichment functions are presented in Section 3.3 and convergence results follow in Section 3.4. Section 3.5 presents real applications where the proposed enrichment functions are used. Finally, some concluding remarks are given in Section 3.6.

3.2 Problem description

Consider in Figure 3.1 an open domain $\Omega \subset \mathbb{R}^2$ with boundary $\Gamma = \overline{\Omega} - \Omega$, the latter having outward unit normal $\mathbf{n}$ and partitioned into mutually exclusive regions Γ_u and Γ_q such that $\Gamma = \overline{\Gamma_u \cup \Gamma_q}$ and $\Gamma_u \cap \Gamma_q = \emptyset$. Dropping the dependance on position $\mathbf{x}$, the strong form for the steady-state thermal boundary value problem can be written as follows: Given the

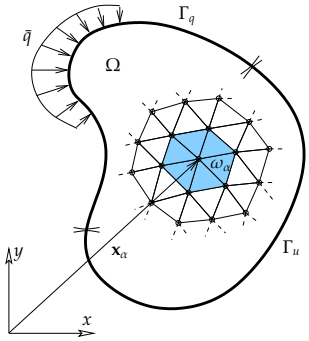


Figure 3.1 – Two-dimensional domain Ω used in the formulation of the problem. The boundary of the domain is split in two mutually exclusive regions Γ_u and Γ_q where Dirichlet and Neumann boundary conditions are applied, respectively. The figure also shows part of a mesh of triangular elements used for discretization showing the cloud or support ω_α for node $\mathbf{x}_\alpha$.

thermal conductivity $\boldsymbol{\kappa} : \overline{\Omega} \to \mathbb{R}^2 \times \mathbb{R}^2$, the heat source $f : \Omega \to \mathbb{R}$, prescribed temperature $\bar{u} : \Gamma_u \to \mathbb{R}$ and prescribed heat flux $\bar{q} : \Gamma_q \to \mathbb{R}$, find the temperature field $u : \overline{\Omega} \to \mathbb{R}$ such that

$$\begin{aligned} \nabla \cdot (\boldsymbol{\kappa} \nabla u) + f &= 0 \quad \text{on } \Omega, \\ u &= \bar{u} \quad \text{on } \Gamma_u, \\ \boldsymbol{\kappa} \nabla u \cdot \mathbf{n} &= \bar{q} \quad \text{on } \Gamma_q. \end{aligned} \tag{3.1}$$

Let $\mathcal{U} = \{u \mid u|_{\Gamma_u} = \bar{u}\} \subset H^1\left(\overline{\Omega}\right)$ be the set of trial solutions for the temperature field and $\mathcal{V} = \{v \mid v|_{\Gamma_u} = 0\} \subset H^1_0\left(\overline{\Omega}\right)$ be the variation space. The weak form of the problem reads: Given $\boldsymbol{\kappa}$, f, $\bar{u}$ and $\bar{q}$ as before, find $u \in \mathcal{U}$ such that

$$a(w, u) = (w, f) + (w, \bar{q})_{\Gamma_q} \qquad \forall w \in \mathcal{V}, \tag{3.2}$$

where the bilinear and linear forms are given by

$$\begin{aligned} a(w,u) &= \int_\Omega \nabla w \cdot (\boldsymbol{\kappa} \nabla u)\, d\Omega, \\ (w,f) &= \int_\Omega w f\, d\Omega, \\ (w,\bar{q})_{\Gamma_q} &= \int_{\Gamma_q} w \bar{q}\, d\Gamma. \end{aligned}$$

For the Galerkin approximation, let $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{U}^h \subset \mathcal{U}$ be finite-dimensional sets such that $\mathcal{V}^h = \left\{ v^h \mid v^h|_{\Gamma_u} = 0 \right\}$ and $\mathcal{U}^h = \left\{ u^h \mid u^h = v^h + \bar{u}^h,\ \bar{u}^h|_{\Gamma_u} \approx \bar{u},\ v^h \in \mathcal{V}^h \right\}$. The Galerkin statement of the boundary value problem is expressed as: Given $\boldsymbol{\kappa}$, f, $\bar{u}$ and $\bar{q}$ as before, find $u^h = v^h + \bar{u}^h \in \mathcal{U}^h$ such that

$$a\left(w^h, v^h\right) = \left(w^h, f\right) + \left(w^h, \bar{q}\right)_{\Gamma_q} - a\left(w^h, \bar{u}^h\right) \qquad \forall w^h \in \mathcal{V}^h. \tag{3.3}$$

The application of Dirichlet boundary conditions within the GFEM framework is not straightforward since some enrichment functions may be non-zero at nodes with a prescribed value of the solution. In this work, the penalty method is adopted due to the simplicity in its implementation. The Galerkin form in this context becomes: Given $\boldsymbol{\kappa}$, f, $\bar{u}$ and $\bar{q}$ as before, find $u^h \in \mathcal{U}^h$ such that

$$a\left(w^h, u^h\right) + \rho\left(w^h, u^h\right)_{\Gamma_u} = \left(w^h, f\right) + \left(w^h, \bar{q}\right)_{\Gamma_q} + \rho\left(w^h, \bar{u}\right)_{\Gamma_u} \qquad \forall w^h \in \mathcal{V}^h, \tag{3.4}$$

where ρ is a penalty parameter. Dirichlet boundary conditions can also be enforced using Lagrange multipliers [77]. A thorough discussion of these and other methods used to enforce Dirichlet boundary conditions can be found in [78].

Let $\Omega^h \equiv \text{int}\left(\cup_{\alpha=1}^{M} \overline{\Omega}_\alpha\right)$ be a discretization of domain Ω in M finite elements such that $\Omega_\alpha \cap \Omega_\beta = \emptyset\ \forall \alpha \neq \beta$. Due to discretization error, $\Omega \cong \Omega^h$ and $\Gamma \cong \Gamma^h \equiv \overline{\Omega^h} - \Omega^h$. Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be the set of N nodes contained in the discretization and $\varphi_\alpha(\mathbf{x})$ be the standard (Lagrangian) finite element shape function associated with node $\mathbf{x}_\alpha$. For this node, let $\omega_\alpha \equiv \{\mathbf{x} \mid \varphi_\alpha(\mathbf{x}) \neq 0\}$ be the *cloud* or *support* of $\mathbf{x}_\alpha$, i.e. the set of all elements attached to it, as illustrated in Figure 3.1 for a mesh of 3-noded triangular elements. The *partition of unity* property of finite element shape functions specifies that

$$\sum_{\alpha=1}^{N} \varphi_\alpha = 1, \qquad \forall \mathbf{x} \in \overline{\Omega^h}. \tag{3.5}$$

In the GFEM, the partition of unity property is used to paste together local enrichment functions $\{L_{\alpha i}(\mathbf{x}) : \omega_\alpha \to \mathbb{R}\}_{i=1}^{E}$ that aim at representing some localized behavior, with E being the number of enrichment functions used in ω_α. In other words,

$$\psi_{\alpha i} = \varphi_\alpha L_{\alpha i} \qquad \text{(no summation on } \alpha\text{)}\,. \tag{3.6}$$

In order to keep the standard finite element shape functions in those elements that contain enriched nodes, we require that $L_{\alpha 0} = 1$, so that a set with E enrichment functions would be $\{1, L_{\alpha i}\}_{i=1}^{E}$. A temperature approximation using the GFEM thus has the form

$$u^h(\mathbf{x}) = \sum_{\alpha=1}^{N} \varphi_\alpha(\mathbf{x})\, \tilde{U}_\alpha + \sum_{\alpha=1}^{N} \varphi_\alpha(\mathbf{x}) \sum_{i=1}^{E} L_{\alpha i}(\mathbf{x})\, \hat{U}_{\alpha i}, \tag{3.7}$$

where the first term corresponds to the standard finite element interpolation and the second term to the enriched/extended part of the approximation, with $\tilde{U}_\alpha$ and $\hat{U}_{\alpha i}$ denoting the standard and enrichment degrees of freedom, respectively. The resulting functions that are used with enriched degrees of freedom can thus be viewed as the cartesian product of the partition of unity shape functions with those of the enrichment set:

$$\{\psi_{\alpha i}\}_{i=0}^{E} = \varphi_\alpha \times \{1, L_{\alpha i}\}_{i=1}^{E}\,. \tag{3.8}$$

Elements where all nodes are enriched are called *reproducing elements* [76]. These are the elements where enrichment functions have to be used in order to capture some localized behavior. With the exception of a few cases, enrichment functions cannot be used directly as in Equation (3.8) because problems arise in those elements that do not have all nodes enriched [79, 76]. These elements, located contiguously to the reproducing elements, are called *blending elements*. Optimal convergence is lost due to pathological terms in blending elements unless the enrichment functions are by construction constant or include polynomial enrichments. A correction to the method recently proposed in [76] will be investigated in this work: Given an enrichment function $\psi_{\alpha i}$, let $\psi^c_{\alpha i}$ be the corrected counterpart, defined as

$$\psi^c_{\alpha i} = \psi_{\alpha i} c, \tag{3.9}$$

where the correction function c over an element is defined as

$$c = \sum_{i \in I^\star} \varphi_i, \tag{3.10}$$

and $I^\star$ is the set of all nodes that belong to reproducing elements in the mesh. In other

words, c is unity in all reproducing elements (due to the partition of unity property), ramps down in blending elements, and is equally zero elsewhere in the domain. The use of this type of *cut-off* function can be traced back to [64], in the context of fracture mechanics. An extension of this correction has been studied in [80] for interacting enrichments. Other approaches have been proposed to overcome the problem that arises in blending elements, including using enhanced formulations [79], hierarchical elements [79, 81] and even using Discontinuous Galerkin (DG) formulations [82].

The choice of some enrichment functions can lead to a singular stiffness matrix. Therefore, the resulting system of equations is positive semidefinite and cannot be solved by standard Gauss elimination or Cholesky decomposition. In this work, the algorithm described in [58, 83] is used, where a solution vector is obtained by carrying out iterative refinement on the solution of a perturbed problem (ill-conditioned but not singular). The perturbation parameter is chosen as $\varepsilon = 10^{-12}$.

3.3 Enrichment functions

This section presents the enrichment functions investigated in this work. The objective is to obtain an enrichment function that is continuous and has a discontinuous gradient in the direction perpendicular to the line segments that represent line loads or grain boundaries. The enrichment functions should be general enough so they can be used in the case of a single interface.

3.3.1 Junction ramp enrichments

Problems where the displacement field is discontinuous (i.e., strong discontinuities) have been addressed in [61, 59, 62]. Most of the enrichment functions for problems with discontinuous field gradients (i.e., weak discontinuities) deal with the case where the discontinuity completely crosses the finite elements [71, 70]. As indicated earlier, for problems with multiple interfaces meeting inside a finite element, enrichment functions based on products of distance functions or Heaviside enrichments have been proposed in [74] and [75], respectively. Here we present other enrichment functions that can be used when dealing with such cases.

Consider the square domain Ω shown in Figure 3.2. The domain is subdivided in regions $\{\mathcal{G}_i\}_{i=1}^{3}$ such that $\overline{\Omega} = \cup_{i=1}^{3}\overline{\mathcal{G}_i}$. Inner sub-domain boundaries are defined as $\Gamma_{ij} \equiv \overline{\mathcal{G}_i \cap \mathcal{G}_j}, i \neq j$ and the junction coordinate as $\mathbf{x}_J = \cap_{i,j=1,i\neq j}^{3}\overline{\Gamma}_{ij}$. In the case of a polycrystalline microstruc-

ture, a region $\mathcal{G}_i$ represents one of the grains in the domain whereas Γ_{ij} represents the material interface between grains i and j. In the case of a homogeneous material, the sub-domain boundaries Γ_{ij} can be viewed as line loads. These line loads become heat sinks in the case of the microvascular material alluded to in Section 3.1.

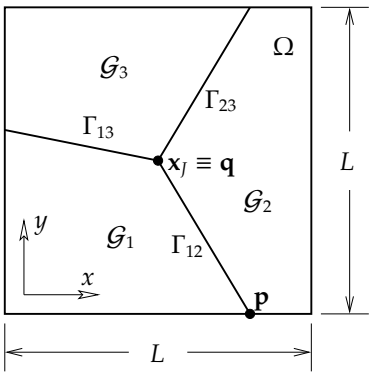


Figure 3.2 – A 2D square domain $\Omega \equiv L \times L$ is divided in regions $\{\mathcal{G}_i\}_{i=1}^{3}$ such that $\overline{\Omega} = \cup_{i=1}^{3} \overline{\mathcal{G}_i}$. Each sub-domain can be viewed as different grains in the case of a polycrystalline microstructure. In the case of a homogeneous material, inner boundaries can be viewed as line loads (heat sinks in the case of a microvascular material for active cooling applications).

The first enrichment considers each sub-domain individually and can be obtained by integrating the enrichment functions proposed for strong discontinuities in polycrystalline materials [59], in the direction perpendicular to the material interfaces (or line loads). For a particular sub-domain, the function ramps only inside it and it is constant elsewhere. For sub-domain $\mathcal{G}_i$, the ramp enrichment function is

$$r_i(\mathbf{x}) = \begin{cases} 1 + \min_{i=1}^{n} d_i(\mathbf{x}) & \text{if } \mathbf{x} \subset \mathcal{G}_i, \\ 1 & \text{otherwise,} \end{cases} \tag{3.11}$$

where $d_i(\mathbf{x})$ is the distance function to the $i-$th line segment representing one of the n inner boundaries of $\mathcal{G}_i$ and the unity constant is introduced so that the resulting matrix is better conditioned. This means that we might need to consider as many enrichment functions as sub-domains. In [59], it was found that for n intersecting inner boundaries, considering $n-1$ enrichments was enough due to the fact that one of the enrichments could be obtained as a linear combination of the others. This issue will be investigated shortly for the type of enrichment functions used in this work. Schematics showing the convention adopted to represent the enrichment function are illustrated in Figure 3.3 for the three sub-domain problem, showing that for each enrichment the function is non-constant in its shaded area.

Arrows in the schematic figures indicate the direction where the ramp function increases in magnitude whereas dashed lines indicate bisector lines between adjacent line segments. The function is C^0 continuous along the bisector lines so they are also considered when subdividing the element for integration purposes, as explained in Section 3.4.3.

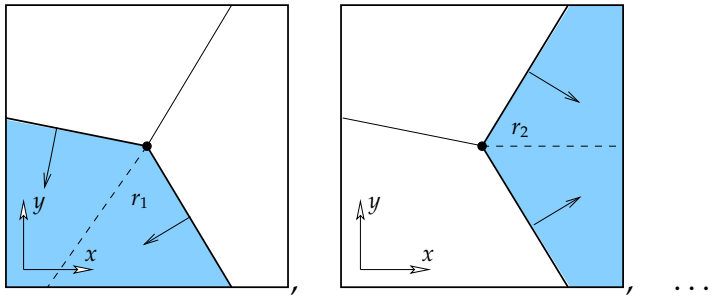


Figure 3.3 – Ramp enrichments given by Equation (3.11), non-constant on their respective shaded area. Arrows denote the direction where the ramps increase in magnitude. Bisector lines are shown as dashed lines.

A second option that involves less degrees of freedom can be obtained by combining the enrichment functions defined in Equation (3.11) into a single junction ramp enrichment $R(\mathbf{x})$. For a junction $\mathbf{x}_J$ between m grains with n boundaries Γ_{ij}, the enrichment function is defined as

$$R(\mathbf{x}) = \sum_{i=1}^{m} r_i(\mathbf{x}) - m = \min_{i=1}^{n} d_i(\mathbf{x}). \tag{3.12}$$

The enrichment function is obtained by computing the distance from point $\mathbf{x}$ to the closest line in the domain. This function is illustrated in Figure 3.4, both in 3D and its equivalent planar representation. Note that, when there is a single line segment (e.g., single material interface), the enrichment function reduces to the enrichment proposed in [70].

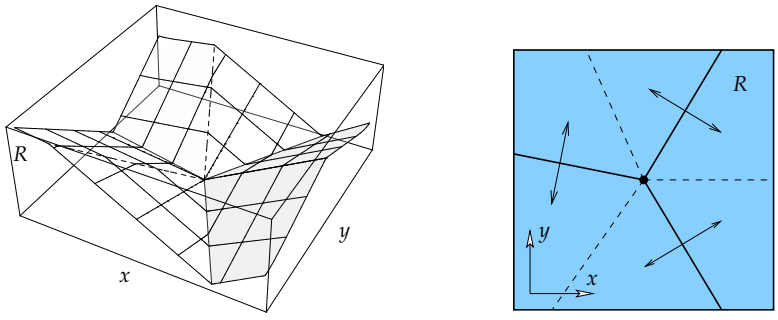


Figure 3.4 – Single junction ramp enrichment $R(\mathbf{x})$ showing a 3D representation (left) and its equivalent 2D schematic (right).

The distance function to the $i-$th line can be computed analytically as follows: the closest point lying on the ray having the same slope as the line segment defined by points $\mathbf{p}$ and $\mathbf{q} \equiv \mathbf{x}_j$ (see Figure 3.2) is $\mathbf{x}^\star = \mathbf{p} + s(\mathbf{x})(\mathbf{q} - \mathbf{p})$, with

$$s(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{p}) \cdot (\mathbf{q} - \mathbf{p})}{\left\|\mathbf{q} - \mathbf{p}\right\|^2}.$$

Then, the distance function is given by

$$d_i(\mathbf{x}) = \begin{cases} \left\|\mathbf{x} - \mathbf{p}\right\| & \text{if } s(\mathbf{x}) \leq 0, \\ \left\|\mathbf{x} - \mathbf{q}\right\| & \text{if } s(\mathbf{x}) \geq 1, \\ \left\|\mathbf{p} + s(\mathbf{x})(\mathbf{q} - \mathbf{p}) - \mathbf{x}\right\| & \text{otherwise.} \end{cases} \tag{3.13}$$

When the closest point on the line segment is obtained through an orthogonal projection (i.e., $0 \leq s(\mathbf{x}) \leq 1$), we approximate the distance function to the $i-$th line segment by using the level set method [84, 74], such that

$$d_i^h(\mathbf{x}) = \left| \sum_j \varphi_j(\mathbf{x}) \Lambda_j^i \right|, \tag{3.14}$$

where Λ_j^i is the level set function value at the $j-$th node of the element that contains point $\mathbf{x}$.

Polynomial functions can also be used together with the ramp functions presented. For a quadratic finite element approximation, we enrich each node $\mathbf{x}_\alpha = (x_\alpha, y_\alpha)$ in the mesh with linear polynomials

$$\xi = \frac{x - x_\alpha}{h_\alpha}, \qquad \eta = \frac{y - y_\alpha}{h_\alpha}, \tag{3.15}$$

where h_α is a scaling parameter related to the size of the cloud of node $\mathbf{x}_\alpha$. The ramp functions are also multiplied by the polynomial enrichments defined in Equation (3.15), so when using, e.g., the single ramp enrichment R the enrichment functions on node $\mathbf{x}_\alpha$ become

$$\mathcal{L} = \{1, \xi, \eta\} \times \{1, R\} = \{1, \xi, \eta, R, \xi R, \eta R\}. \tag{3.16}$$

These high-order enrichment functions follow the same concept as those proposed in [85, 86].

The aforementioned enrichment functions r_i and R defined in Equations (3.11) and (3.12), respectively, are used to enrich all nodes whose support intersect any of their subdomain boundaries. However, there are situations where the required information to

produce these enrichments is not easily available. The information about the intersecting lines can be obtained locally when evaluating the junction even if the complete geometric description of the sub-domains is not available. The proposed enrichments can still be applied to the finite element nodes with support interacting with the junction. Other nodes of elements completely split by the interfaces can be enriched with other interface enrichments, thus allowing the mixing of the enrichments proposed here with other enrichment functions available in the literature. Figure 3.5 shows schematically the mixing between interface and junction enrichments. The nodes of elements that are completely cut by the sub-domain inner boundaries are enriched with interface enrichments unless they already have a junction enrichment (see Figure 3.5b). The convention used to denote the set of enrichment functions used at a node when mixing types is

$$\mathcal{L} = \left\{1, \left[\left\{L^I_{\alpha i}(\mathbf{x})\right\}_{i=1}^{E_I} \mid \left\{L^J_{\alpha i}(\mathbf{x})\right\}_{i=1}^{E_J}\right]\right\}, \tag{3.17}$$

where E_I and E_J represent the number of enrichment functions used for *interfaces* and *junctions*, respectively, and $[\cdot|\cdot]$ denotes one set of enrichments or the other, but not both. For example, the enrichment set $\mathcal{L} = \{1, \xi, \eta\} \times \{1, [M|R]\}$ is equivalent to

$$\mathcal{L} = \left\{1, \xi, \eta, \left[\left\{L^I_{\alpha i}\right\}_{i=1}^{3} \mid \left\{L^J_{\alpha i}\right\}_{i=1}^{3}\right]\right\} = \{1, \xi, \eta, [M, M\xi, M\eta \mid R, R\xi, R\eta]\},$$

where $\left\{L^I_{\alpha i}\right\}_{i=1}^{3} = \{M, M\xi, M\eta\}$ and $\left\{L^J_{\alpha i}\right\}_{i=1}^{3} = \{R, R\xi, R\eta\}$ denote the sets of enrichment functions used for interfaces and junctions, respectively. The nodes of the element that contains the junction are enriched with the functions described above. As a result, the junction function is non-zero over $\Omega^L \equiv \omega_\alpha \cup \omega_\beta \cup \omega_\gamma$. Thus, we use a unique enrichment function that ramps in all directions inside Ω^L (Figure 3.6b) or single ramp enrichments for each pair of adjacent lines considering them as part of a fictitious sub-domain (Figure 3.6a). This approach also allows us to combine the junction enrichments proposed above with other interface enrichments available in the literature.

The correction proposed by Fries [76] may be used in conjunction with the proposed enrichments. The corrected enrichment functions require an additional layer of nodes to be enriched, thus increasing the size of the support of the function and consequently the number of degrees of freedom. In other words, not only the nodes of elements that interact with inner boundaries are enriched but also those of contiguous elements (i.e., blending elements). The functions, taking into account the correction, are denoted hereafter as r_i^c and R^c.

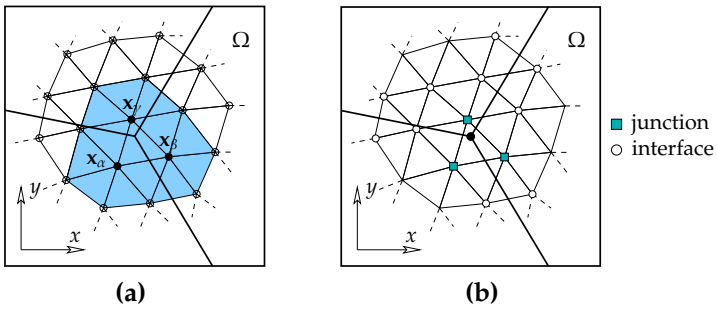


Figure 3.5 – Mixing between interface and junction enrichments. (a) Area $\Omega^L \equiv \omega_\alpha \cup \omega_\beta \cup \omega_\gamma$ (shaded area) where the junction function is applied. (b) Choice of interface and junction enrichments.

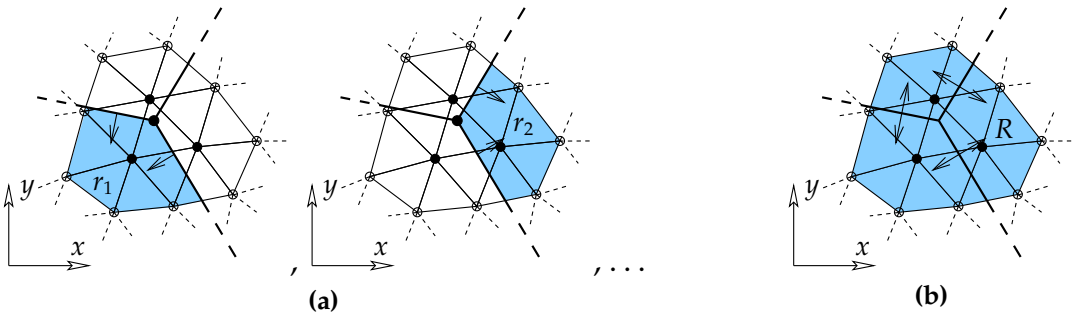


Figure 3.6 – Junction enrichment functions applied only to those nodes with support interacting with the junction $\mathbf{x}_J$. This approach is appealing for problems where the complete geometry of the sub-domains cannot be defined easily or when mixing enrichment types. (a) $r_1, r_2, \ldots \forall \mathbf{x} \in \Omega^L$; (b) R $\forall \mathbf{x} \in \Omega^L$.

3.3.2 Interface enrichments

Interface enrichments have been investigated primarily for inclusions and voids inside another material. Even though many examples in this work do not have a material interface per se, these functions are still referred to as *interface enrichments* to be consistent with the existing literature. These functions can be used to enrich the nodes of elements that are completely cut by the lines, unless they are enriched with junction ramp enrichments r_i or R or their corrected counterparts.

Ridge enrichment

Introduced by Möes et al. [71], the ridge enrichment function is given by

$$M(\mathbf{x}) = \sum_i^N |\Lambda_i| \varphi_i(\mathbf{x}) - \left| \sum_i^N \Lambda_i \varphi_i(\mathbf{x}) \right|, \tag{3.18}$$

where Λ_i is the level set function value corresponding to the $i-$th node. This function constructs a continuous field inside the element with a ridge following the path of the interface. By construction, this enrichment function is identically zero in elements that do not contain the interface. As a result, no problems arise in blending elements and the correction given by Equation (3.10) is not necessary.

Ramp enrichments

These functions are a special case of the ramp enrichments presented before when considering a single line segment. The function R ramps in both perpendicular directions from the line segment that represents the inner boundary. This function was proposed by Sukumar et al. in [70] with special treatment on blending elements. Chessa et al. made use of this function for solidification problems [87]. Fries [76] used this function together with his proposed correction and showed optimal convergence rates in the case of a circular inclusion for 2D elasto-statics. In this work the latter is denoted R^c. Similarly, the functions $r_i, i = 1, 2$ and their corrected counterparts introduced before can be used for interface enrichments when considering a single line segment. These enrichment functions ramp to one side of the line segment and are constant on the other side. A single ramp function on one side of the interface was used in [88] for a comparison between the XFEM and the Immerse Interface Method.

3.4 Convergence results

Convergence results for all enrichment functions investigated are presented in this section.

3.4.1 Single uniform heat source

This example is used to introduce the enrichment functions used in this work in the context of a single line load. Let the temperature field over $\Omega \equiv L \times L$ (Figure 3.7) be defined as

$$u(x,y) = \begin{cases} \frac{x(L-4x)(5L-4x)(L-2x)}{6L^3} & x \leq L/2, \\ \frac{(3L-4x)(L-2x)(L-x)(L+4x)}{6L^3} & x \geq L/2. \end{cases} \tag{3.19}$$

This function, which is constant in the y direction, was manufactured from two polynomials $X_1(x)$ and $X_2(x)$ at each side of the line $x = L/2$ such that $[\![u'(L/2)]\!] = -1$. This constant jump along the line $x = L/2$ can be clearly seen in Figure 3.8. The body heat source term

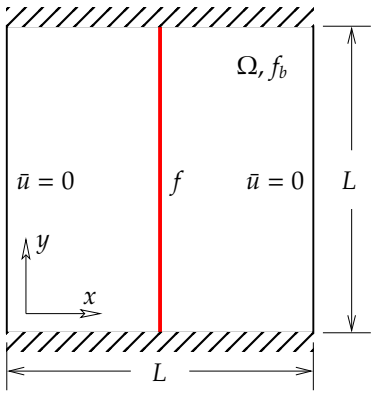


Figure 3.7 – Schematic for the single uniform line heat source. A 2D square domain $\Omega = L \times L$ contains a single line heat source f that traverses it from side to side and a body heat source f_b. Boundary conditions include prescribed temperature $u = 0$ at left and right edges, and insulated bottom and top edges.

that needs to be applied to all elements results from substituting Equation (3.19) in the differential equation:

$$f_b(x,y) = \frac{-102L^4 + 576L^3x - 576L^2x^2}{9L^5}. \tag{3.20}$$

For the finite element solutions, the domain Ω^h is then discretized using matching (M) and non-matching (NM) meshes. A single line heat source of unit magnitude per unit length

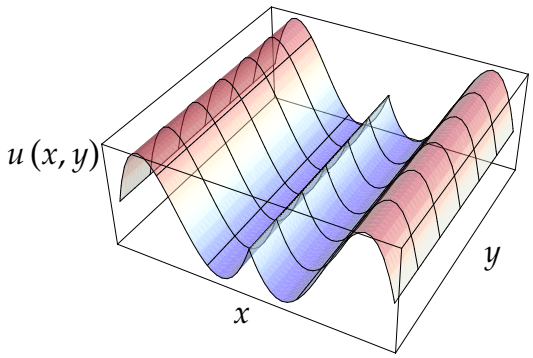


Figure 3.8 – Temperature distribution given by Equation (3.19) showing the constant jump in the derivative at $x = L/2$.

traverses the domain. This line load is the responsible for creating the jump $[\![u'(L/2)]\!] = -1$. Recall that by a matching mesh we mean that the heat source follows the edges of the finite elements. Top and bottom edges are insulated (i.e., $\bar{q} = 0$) and a temperature $\bar{u} = 0$ is prescribed at left and right edges. The Dirichlet boundary conditions are enforced using the penalty method because the example involves the use of the polynomial enrichments given by Equation (3.15).

The results from the convergence study for this problem are illustrated in Figures 3.9 and 3.10 for the L_2 and energy norms, respectively. Figures on the top show in abscissas the number of degrees of freedom n whereas the figures on the bottom show the mesh size h. The error in the L_2 is given by

$$\left\|u - u^h\right\|_{L_2(\Omega)} \equiv \sqrt{\int_\Omega (u - u^h)^2 \, d\Omega},$$

whereas the error in the energy norm is

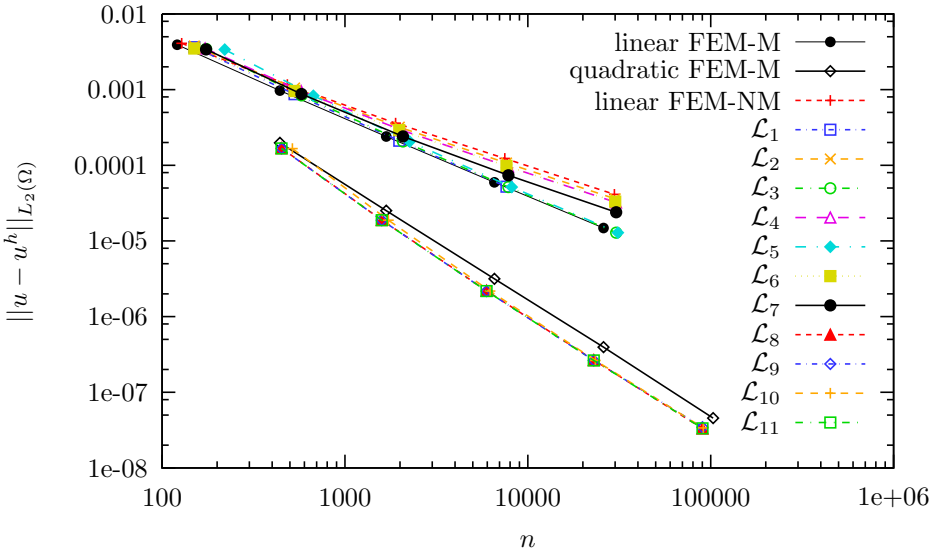
$$\left\|u - u^h\right\|_{E(\Omega)} \equiv \sqrt{\left\|u - u^h\right\|^2_{L_2(\Omega)} + \int_\Omega \left\|\nabla u - \nabla u^h\right\|^2 d\Omega}.$$

Our reference solutions are the standard finite element solutions on matching meshes, denoted as FEM-M in Figures 3.9 and 3.10. The linear FEM-M, obtained with standard 3-noded elements, attains optimal convergence of 2 in the L_2 norm and 1 in the energy norm with respect to the mesh size h. 6-noded elements are used in the quadratic FEM-M, and optimal convergence rates of 3 and 2 are obtained. Refer to Table 3.1 for a complete list of convergence values. The FEM-NM solutions refer to those of non-matching meshes without the use of enrichment functions. The purpose of showing these solutions is two-fold:

	$h-E$	$h-L_2$	$n-E$	$n-L_2$
linear FEM-M	1	2	0.5	1
quadratic FEM-M	2	3	1	1.51
linear FEM-NM	0.54	1.58	0.27	0.79
$\mathcal{L}_1 = \{1, M\}$	1	2.02	0.51	1.04
$\mathcal{L}_2 = \{1, r_1\}$	0.53	1.57	0.27	0.8
$\mathcal{L}_3 = \left\{1, r_1^c\right\}$	1	2	0.51	1.02
$\mathcal{L}_4 = \{1, r_1, r_2\}$	0.53	1.58	0.27	0.8
$\mathcal{L}_5 = \left\{1, r_1^c, r_2^c\right\}$	0.99	2	0.51	1.04
$\mathcal{L}_6 = \{1, R\}$	0.53	1.57	0.27	0.79
$\mathcal{L}_7 = \{1, R^c\}$	0.56	1.62	0.29	0.83
$\mathcal{L}_8 = \{1, \xi, \eta\} \times \{1, M\}$	2	3	1.02	1.52
$\mathcal{L}_9 = \{1, \xi, \eta\} \times \{1, r_1\}$	1.93	2.93	0.98	1.49
$\mathcal{L}_{10} = \{1, \xi, \eta\} \times \{1, r_1, r_2\}$	1.97	2.96	1	1.51
$\mathcal{L}_{11} = \{1, \xi, \eta\} \times \{1, R\}$	2	2.98	1.01	1.51

Table 3.1 – Convergence rates for the single uniform heat source example. All convergence rates reported are obtained using the two most refined solutions.

Firstly, this solution establishes an upper bound on the error of other solutions. Secondly, even though the convergence rate is very poor when compared to other solutions, the standard FEM still converges as the meshes are refined because the interface is contained in increasingly smaller elements. The curve $\mathcal{L}_1 = \{1, M\}$ corresponds to the use of the ridge function proposed in [71], showing optimal performance for this problem. All ramp enrichment functions without the correction proposed in [76] perform as poorly as the FEM-NM. Adding the correction enables them to recover optimal convergence rates for all ramp functions but the one that ramps to both sides of the interface (i.e., R^c). For the quadratic approximations using enrichments, all nodes in the mesh are enriched with the linear polynomials given by Equation (3.15). As explained before, the discontinuous part of the approximation is also multiplied by polynomials so the enrichment functions are given by Equation (3.16). The results show that quadratic optimal convergence is obtained in all cases, without the use of Fries' correction term and regardless of the enrichment used. Note also that all approximations using enrichment functions are more accurate than the quadratic standard FE approximation on matching meshes. Furthermore, for any given number of degrees of freedom n, quadratic approximations are more accurate than the linear approximations. Therefore, quadratic approximations should be preferred over linear ones, even for very coarse finite element meshes when solving this class of problems.



$$
\begin{array}{lcllcllcl}
\mathcal{L}_1 & = & \{1, M\}\,, & \mathcal{L}_2 & = & \{1, r_1\}\,, & \mathcal{L}_3 & = & \left\{1, r_1^c\right\}, \\
\mathcal{L}_4 & = & \{1, r_1, r_2\}\,, & \mathcal{L}_5 & = & \left\{1, r_1^c, r_2^c\right\}, & \mathcal{L}_6 & = & \{1, R\}\,, \\
\mathcal{L}_7 & = & \{1, R^c\}\,, & \mathcal{L}_8 & = & \{1, \xi, \eta\} \times \{1, M\}\,, & \mathcal{L}_9 & = & \{1, \xi, \eta\} \times \{1, r_1\}\,, \\
\mathcal{L}_{10} & = & \{1, \xi, \eta\} \times \{1, r_1, r_2\}\,, & \mathcal{L}_{11} & = & \{1, \xi, \eta\} \times \{1, R\}\,, &&&
\end{array}
$$

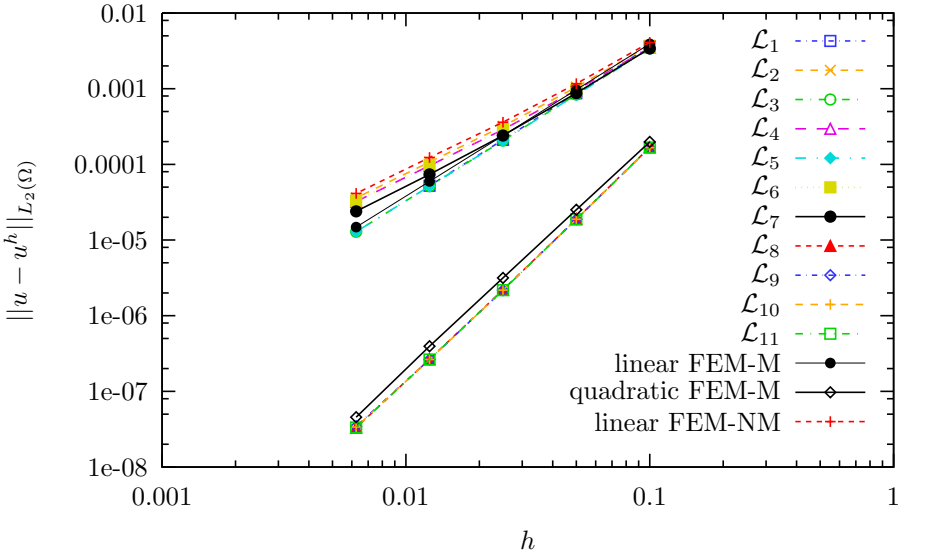
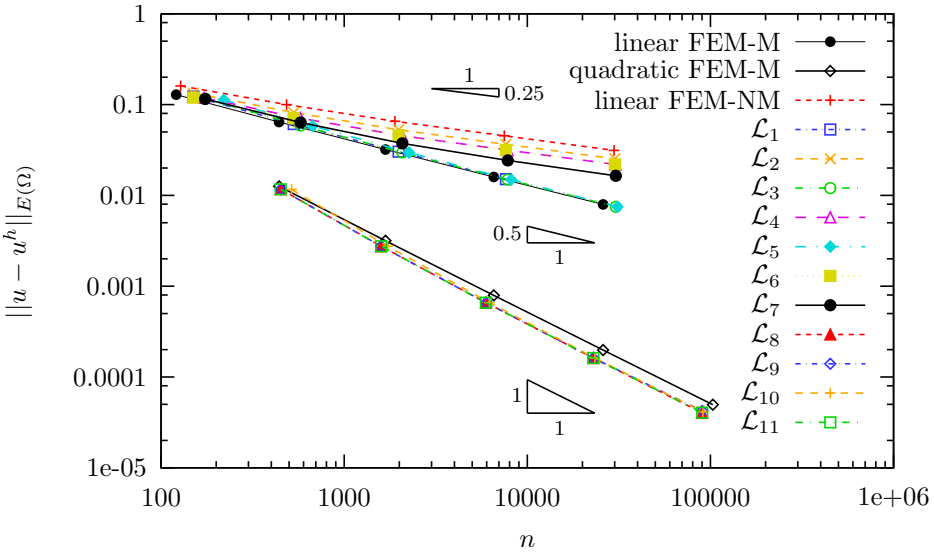


Figure 3.9 – Convergence results in the L_2 norm for the single uniform heat source example shown in Figure 3.7. The top figure shows in abscissas the number of degrees of freedom n whereas the bottom figure shows the mesh size h.



$$
\begin{aligned}
\mathcal{L}_1 &= \{1, M\}, & \mathcal{L}_2 &= \{1, r_1\}, & \mathcal{L}_3 &= \left\{1, r_1^c\right\}, \\
\mathcal{L}_4 &= \{1, r_1, r_2\}, & \mathcal{L}_5 &= \left\{1, r_1^c, r_2^c\right\}, & \mathcal{L}_6 &= \{1, R\}, \\
\mathcal{L}_7 &= \{1, R^c\}, & \mathcal{L}_8 &= \{1, \xi, \eta\} \times \{1, M\}, & \mathcal{L}_9 &= \{1, \xi, \eta\} \times \{1, r_1\}, \\
\mathcal{L}_{10} &= \{1, \xi, \eta\} \times \{1, r_1, r_2\}, & \mathcal{L}_{11} &= \{1, \xi, \eta\} \times \{1, R\},
\end{aligned}
$$

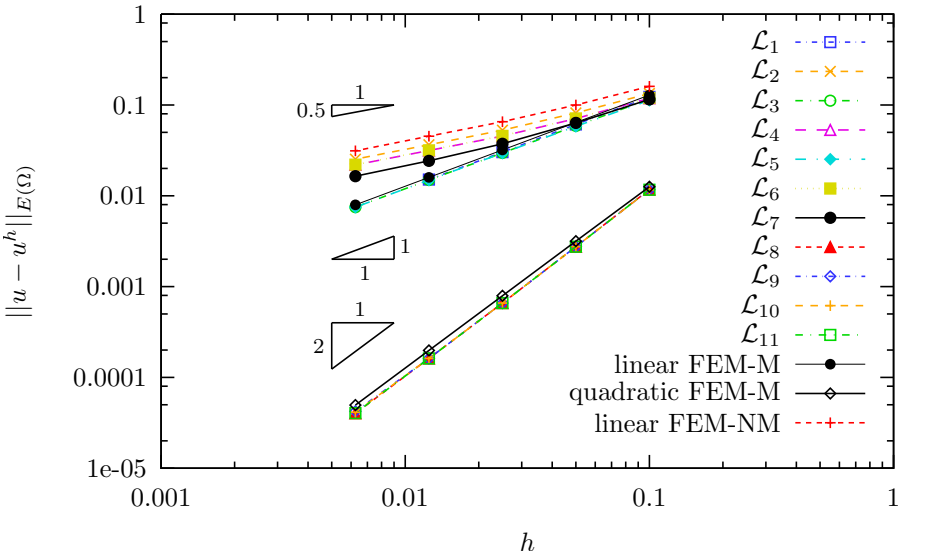


Figure 3.10 – Convergence results in the energy norm for the single uniform heat source example shown in Figure 3.7. The top figure shows in abscissas the number of degrees of freedom n whereas the bottom figure shows the mesh size h.

3.4.2 Single linearly varying heat source

There might be cases where the jump in the derivative of the solution is not constant but spatially varying. To investigate this situation we multiply the temperature field from the previous example by $Y(y) = y - L/2$, which results in a field that has a linearly varying jump in the derivative along the line load (i.e., $[\![u_{,x}(L/2, y)]\!] = -Y(y)$). The temperature field is then

$$u(x,y) = \begin{cases} \frac{x(L-4x)(5L-4x)(L-2x)(2y-L)}{12L^3} & x \leq L/2, \\ \frac{(3L-4x)(L-2x)(L-x)(L+4x)(2y-L)}{12L^3} & x \geq L/2. \end{cases} \tag{3.21}$$

and again it is shown in Figure 3.11. The body source applied to all elements in the

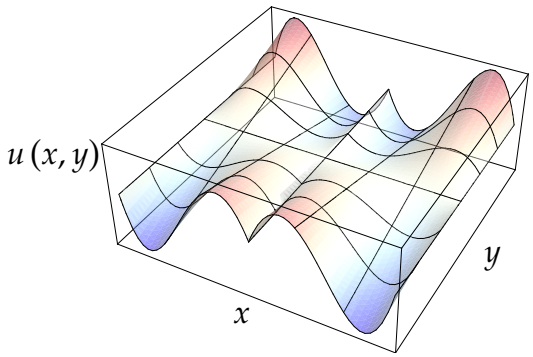
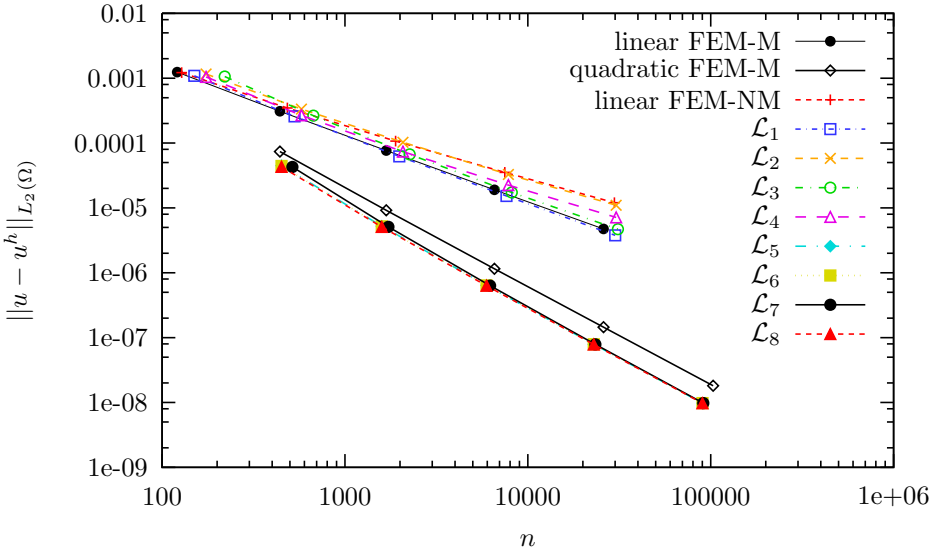


Figure 3.11 – Temperature distribution given by Equation (3.21) showing the linear jump in the derivative at $x = L/2$.

discretization is the same as that given by Equation (3.20). Left and right edges are Dirichlet boundaries again using the penalty method. Top and bottom edges this time have a prescribed heat flux ($\bar{q} = u_{,y}$ for the top edge and $\bar{q} = -u_{,y}$ for the bottom edge).

The convergence results for this example are shown in Figures 3.12 and 3.13, where non-optimal enrichment functions from the previous example were excluded. The results show that the corrected ramp enrichments used previously do not achieve optimal convergence anymore. Therefore, the corrected ramp functions can only represent a constant jump, which is clear because the nature of the function. In other words, the function has along the line a constant jump in the derivative normal to the line so a linear variation is not possible. However, using the ramp functions with linear polynomial enrichments brings optimal convergence back, as shown in the figure for the quadratic approximations. Interestingly, the ridge enrichment function performs as well as before. This ridge function, which is not constant over the ridge, performs better than the ramp functions when the jump is not constant. Furthermore, no correction is needed since the function is already zero in blending elements. As a result, this function uses less degrees of free-



$$\mathcal{L}_1 = \{1, M\}, \quad \mathcal{L}_2 = \left\{1, r_1^c\right\}, \quad \mathcal{L}_3 = \left\{1, r_1^c, r_2^c\right\},$$
$$\mathcal{L}_4 = \{1, R^c\}, \quad \mathcal{L}_5 = \{1, \xi, \eta\} \times \{1, M\}, \quad \mathcal{L}_6 = \{1, \xi, \eta\} \times \{1, r_1\},$$
$$\mathcal{L}_7 = \{1, \xi, \eta\} \times \{1, r_1, r_2\}, \quad \mathcal{L}_8 = \{1, \xi, \eta\} \times \{1, R\},$$

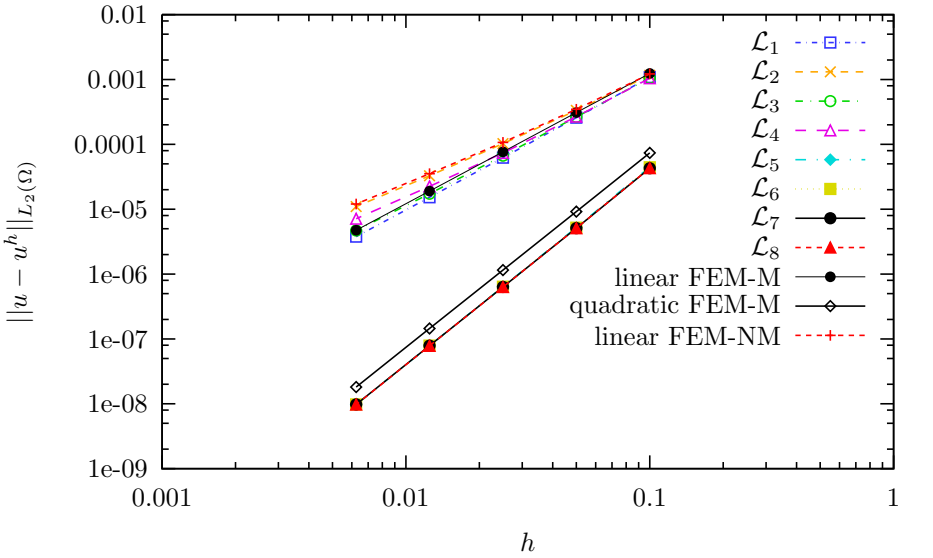
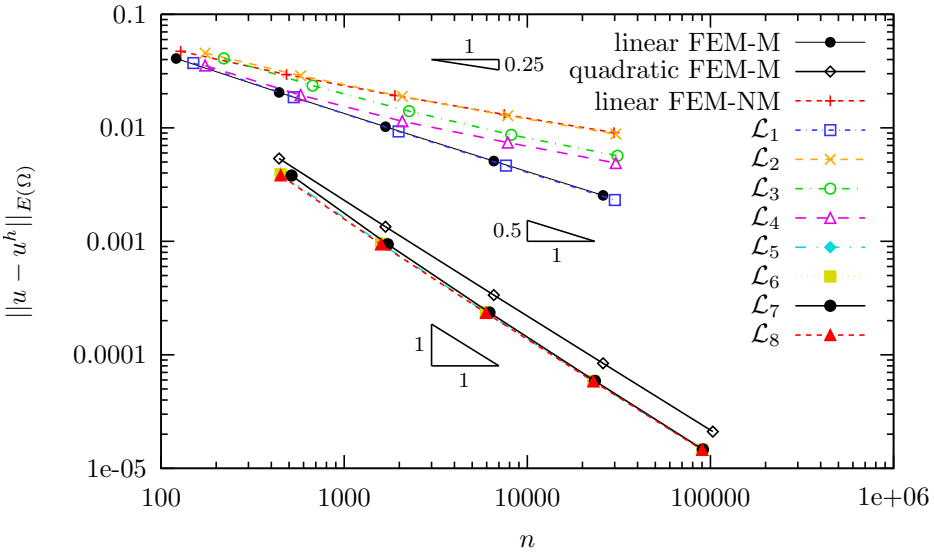


Figure 3.12 – Convergence results in the L_2 norm for the linearly varying heat source example. Top and bottom figures show in abscissas the number of degrees of freedom n and the mesh size h, respectively.



$$\mathcal{L}_1 = \{1, M\}, \quad \mathcal{L}_2 = \left\{1, r_1^c\right\}, \quad \mathcal{L}_3 = \left\{1, r_1^c, r_2^c\right\},$$
$$\mathcal{L}_4 = \{1, R^c\}, \quad \mathcal{L}_5 = \{1, \xi, \eta\} \times \{1, M\}, \quad \mathcal{L}_6 = \{1, \xi, \eta\} \times \{1, r_1\},$$
$$\mathcal{L}_7 = \{1, \xi, \eta\} \times \{1, r_1, r_2\}, \quad \mathcal{L}_8 = \{1, \xi, \eta\} \times \{1, R\},$$

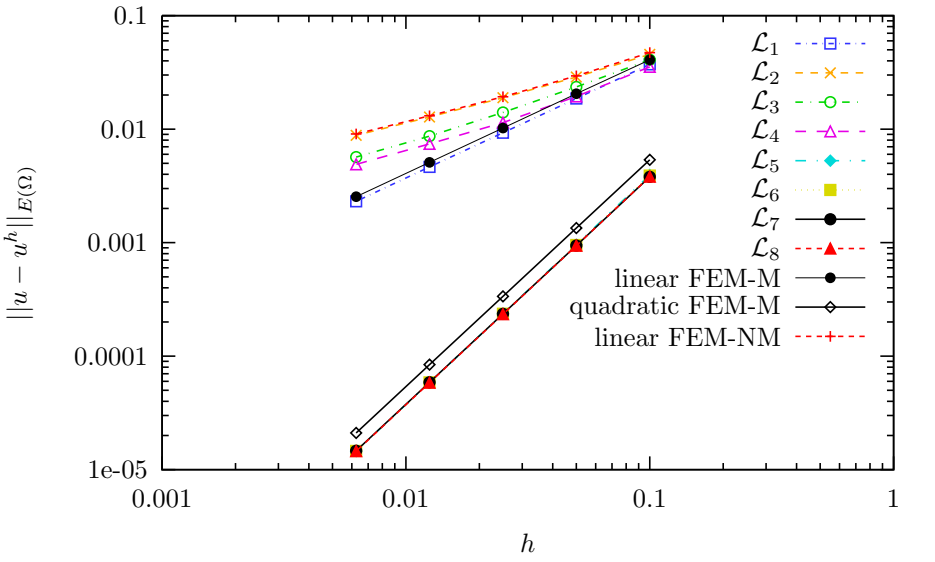


Figure 3.13 – Convergence results in the energy norm for the linearly varying heat source example. Top and bottom figures show in abscissas the number of degrees of freedom n and the mesh size h, respectively.

	$h-E$	$h-L_2$	$n-E$	$n-L_2$
linear FEM-M	1	2.01	0.51	1.01
quadratic FEM-M	2	3	1	1.5
linear FEM-NM	0.53	1.55	0.27	0.78
$\mathcal{L}_1 = \{1, M\}$	1	2.02	0.51	1.02
$\mathcal{L}_2 = \left\{1, r_1^c\right\}$	0.53	1.58	0.27	0.81
$\mathcal{L}_3 = \left\{1, r_1^c, r_2^c\right\}$	0.61	1.88	0.32	0.98
$\mathcal{L}_4 = \{1, R^c\}$	0.6	1.66	0.31	085
$\mathcal{L}_5 = \{1, \xi, \eta\} \times \{1, M\}$	2.01	3.01	1.02	1.53
$\mathcal{L}_6 = \{1, \xi, \eta\} \times \{1, r_1\}$	2.01	3.01	1.02	1.53
$\mathcal{L}_7 = \{1, \xi, \eta\} \times \{1, r_1, r_2\}$	2	3.01	1.02	1.53
$\mathcal{L}_8 = \{1, \xi, \eta\} \times \{1, R\}$	2.01	3.01	1.02	1.52

Table 3.2 – Convergence rates for the linearly varying heat source example. All convergence rates reported are obtained using the two most refined solutions.

dom than the ramps for linear approximations and should be used when possible. For quadratic approximations, the ramp functions perform as well as the ridge function, and they have the same number of degrees of freedom (except when using two single-sided ramps). Table 3.2 lists all convergence rates obtained in this example.

3.4.3 Multiple uniform heat sources

The same domain Ω used in the previous examples is used here. The problem now contains three line heat sources of unit magnitude per unit length that meet at the center of the domain, as illustrated in Figure 3.14. An exact solution for this problem is not available so convergence rates will be measured with respect to an energy value obtained using a cubic approximation on a very fine matching mesh (10-node triangles). The error in the energy norm is computed as

$$\left\|u - u^h\right\|_{E(\Omega)} \equiv \sqrt{a\left(u, u\right) - a\left(u^h, u^h\right)}. \tag{3.22}$$

The boundary conditions for this example include prescribed temperature along the right edge whereas all other edges are insulated.

Figure 3.15 shows typical integration meshes used in this example when neglecting and considering bisector lines. In the latter case, the bisector lines are considered in addition to the lines that define the heat sources for the initial element subdivision into integration elements. Thus, the resulting partitioned elements that lie along bisector directions are forced to have their edges aligned with the bisector lines. During the computation of

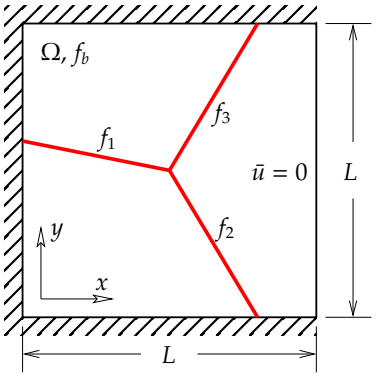
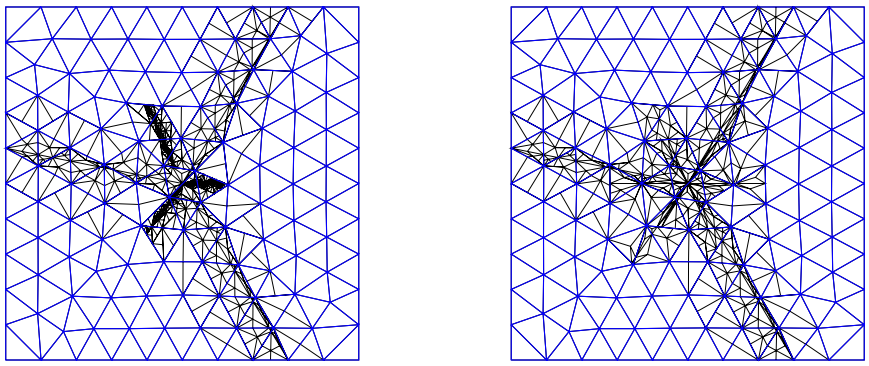


Figure 3.14 – Geometry of test problem with multiple uniform line heat sources. A 2D square domain Ω of side L contains uniform heat sources $f_i, (i = 1, 2, 3)$ that meet at the center of the domain. Boundary conditions include prescribed temperature $u = 0$ at the right edge whereas the remaining edges are insulated.

local matrices and vectors, elements are created in places where the functions are not smooth and therefore difficult to integrate. This can be seen in Figure 3.15a, where not considering bisector lines for the initial partitioning has created smaller integration elements along the bisector directions. Thus, this technique can be used together with the GFEM to find regions of difficult integration inside the mesh. The algorithm used in this work uses adaptive integration only in elements with enriched nodes. No adaptive integration is used in the rest of the mesh since in those elements a standard finite element approximation is used (as long as no polynomial enrichments are used). Moreover, a single level of recursion is used in blending elements because the enrichment function in those elements is smooth. Adding bisector lines reduces dramatically the level of recursion used in elements that lie in the path of bisector lines, as noted in Figure 3.15b. The use of this technique within the GFEM framework is not new and can be traced back to [89, 69].

Convergence results for this problem are summarized in Figure 3.16. As in the previous examples, FEM-M and FEM-NM denote the standard finite element solutions on matching and non-matching meshes, respectively. The figure shows that the second junction enrichment proposed (i.e., R^c) performs sub-optimally for a linear approximation. Having a unique function does not provide enough degrees of freedom to represent the solution accurately. Adding polynomial enrichments improves dramatically the enrichment, but the solutions are still not as accurate as other solutions with the same polynomial degree that we will discuss shortly. The following curves show the results of using the first proposed enrichment for junctions (i.e., considering all subdomains with individual enrichments).



(a) (b)

Figure 3.15 – Resulting integration mesh using adaptive numerical quadrature. (a) Bisector lines not considered; (b) Bisector lines considered.

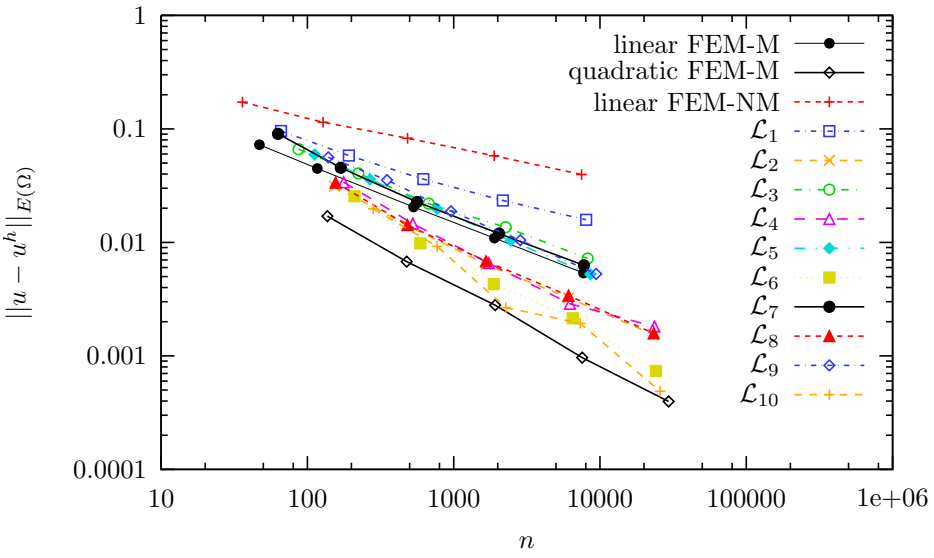
It can be seen from the results that considering three enrichment functions gives more accurate results than using only two. This is in contrast to the results found in [59] for discontinuous displacement fields in polycrystalline materials, where $n-1$ enrichments are used on a junction of n grains because one of the enrichments was linearly dependent. Here the functions within the grains are completely different among them so the solutions obtained considering n enrichments for an $n-$junction gives more accurate results. Finally, taking the ridge function proposed in [71] as the choice for interfaces, we use local junction enrichments. Once again we see that using a unique function to represent the junction does not perform as well as having individual enrichments, even though it requires less degrees of freedom. Convergence rates are not reported for this example because the reference energy has a finite accuracy. This reference value is accurate enough for the linear approximations, but it affects the rates for the quadratic ones.

3.5 Applications

This section presents two real applications where the proposed enrichment functions are used.

3.5.1 Actively cooled microvascular material

The goal of this example is to analyze a microvascular material with active cooling capabilities, thus using the proposed enrichments in a real problem. Much work has been



$$
\begin{aligned}
\mathcal{L}_1 &= \{1, R^c\}, & \mathcal{L}_2 &= \{1, \xi, \eta\} \times \{1, R\}, \\
\mathcal{L}_3 &= \left\{1, r_1^c, r_2^c\right\}, & \mathcal{L}_4 &= \{1, \xi, \eta\} \times \{1, r_1, r_2\}, \\
\mathcal{L}_5 &= \left\{1, r_1^c, r_2^c, r_3^c\right\}, & \mathcal{L}_6 &= \{1, \xi, \eta\} \times \{1, r_1, r_2, r_3\}, \\
\mathcal{L}_7 &= \{1, [M|R^c]\}, & \mathcal{L}_8 &= \{1, \xi, \eta\} \times \{1, [M|R]\}, \\
\mathcal{L}_9 &= \left\{1, \left[M|r_1^c, r_2^c, r_3^c\right]\right\}, & \mathcal{L}_{10} &= \{1, \xi, \eta\} \times \{1, [M|r_1, r_2, r_3]\},
\end{aligned}
$$

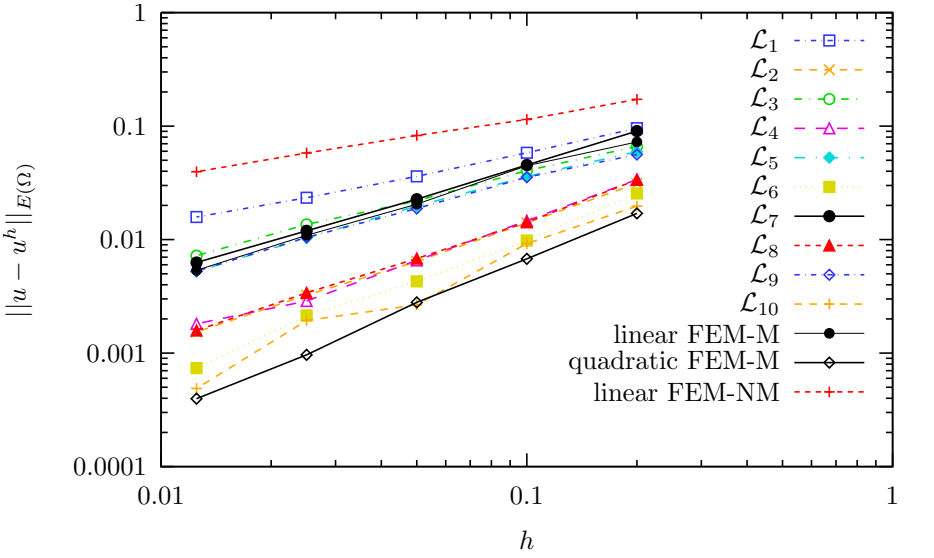


Figure 3.16 – Convergence results in the energy norm for the multiple heat sources example. The top figure shows in abscissas the number of degrees of freedom n whereas the bottom figure shows the mesh size h.

done on the development of materials that mimic living organisms to provide self-healing and active cooling capabilities [7, 8, 31]. Chapter 2 presents a detailed discussion on the optimization of microvascular materials to void volume fraction and flow efficiency. For an active cooling microvascular material, thermal loads are applied and an embedded network containing a cooling fluid is used to reduce its maximum temperature. It can be shown (see reference [90]) that the equivalent heat sink generated by the fluid in a single microchannel is given by

$$q_e = \dot{m}_e c_p \frac{du}{d\xi_e}, \tag{3.23}$$

where $\dot{m}_e$ and c_p are the the mass flow rate in the microchannel and the specific heat of the fluid, respectively, and ξ_e is the local coordinate in the direction of the channel. This simplified formulation for the *conjugate heat transfer* problem, whose assumptions lie outside the scope of this work, allows us to collapse the cooling effect of the fluid to a heat sink applied over the centerlines of the channels. The dependance of the heat sink on the temperature field implies that Equation (3.23) will contribute to the stiffness matrix in the finite element formulation. Of course, the accuracy in the representation of these sink terms is limited to the type of approximation used. In other words, the formulation above can represent a linear variation on the heat sink if a quadratic approximation is used. Furthermore, the stiffness matrix loses its symmetry due to the addition of these sink terms. The algorithm described in [58, 83] can still be used to solve the resulting system of equations with small changes.

Consider the mathematical model of a biomimetic active cooling material depicted in Figure 3.17. The material is composed of epoxy with thermal conductivity $\kappa = 0.3\,\mathrm{W/mK}$. The material is represented by the rectangular domain Ω with sides $L_x = 66\,\mathrm{mm}$ and $L_y = 68\,\mathrm{mm}$. The microvascular cooling network, denoted as $\mathcal{G}$ in the figure, contains microchannels that follow Murray's law [91]: At any junction,

$$\sum_i D_{pi}^3 = \sum_j D_{cj}^3,$$

where D_p and D_c denote the diameters of microchannels with inflow to and outflow from the junction, respectively. Murray's law ensures that the distribution of diameters in the network minimizes the pressure drop between the inlet and the outlet locations, given that the flow inside the microchannels is laminar. Innermost microchannels to the x barycentric axis have the smallest diameter $D = 200\,\mu\mathrm{m}$.

To determine the pressure distribution in the network, the Hagen-Poiseuille law is used

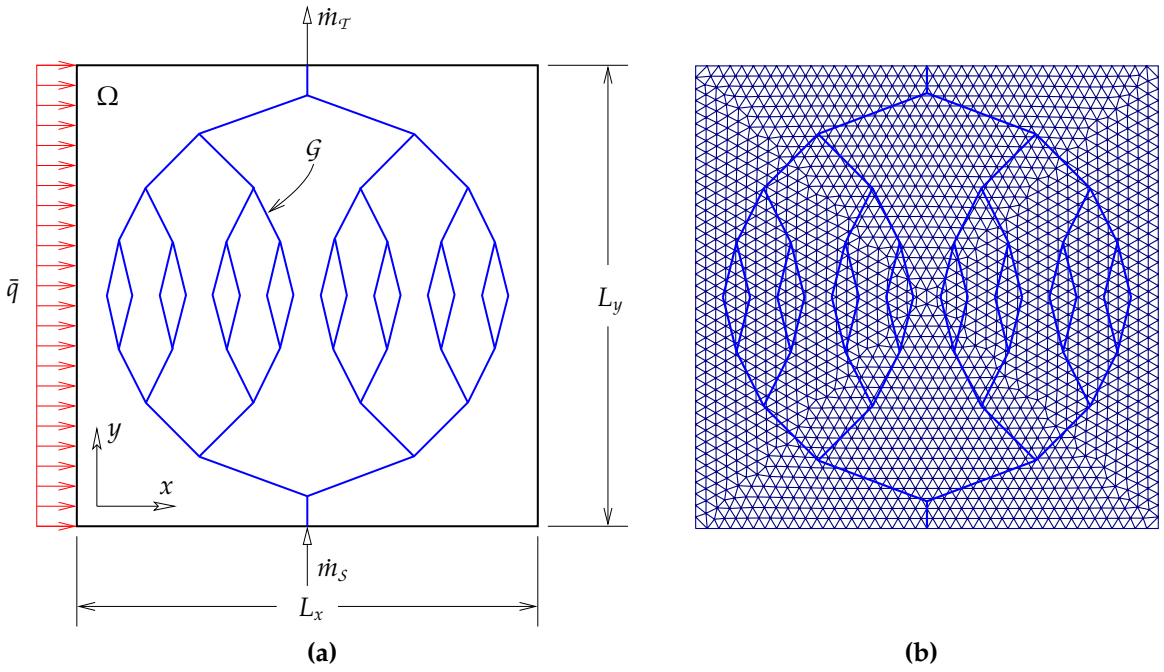


Figure 3.17 – (a) Mathematical model of a biomimetic active cooling material composed of epoxy. The material has en embedded microvascular network with a distribution of diameter values that follows Murray's law. The domain contains a single inflow $\dot{m}_S$ and a single outflow $\dot{m}_T$ located on the bottom and top edges, respectively. A prescribed heat flux $\bar{q}$ is applied to the left edge whereas the remaining edges are convective boundaries. (b) Non-conforming finite element mesh used for the discretization.

to represent the pressure drop in the microchannel:

$$\Delta p_e = \frac{128 \nu L_e}{\pi D_e^4} \dot{m}_e, \tag{3.24}$$

where L_e and D_e denote the length and diameter of the microchannel e, respectively, and ν the kinematic viscosity of the fluid. Assembling the contribution of all microchannels in the network results in a linear system of equation $\vec{K}\vec{p} = \vec{m}$, where $\vec{K}$ is the characteristic matrix (the equivalent to the stiffness matrix in solid mechanics), $\vec{p}$ is the pressure vector and $\vec{m}$ is the mass flow rate vector. The boundary conditions consist of a prescribed water mass inflow of 20 g/min and prescribed atmospheric pressure at the outflow.

The boundary value problem for the temperature solution has a prescribed heat flux $\bar{q} = 500\,\mathrm{W/m}$ on the left edge. The remaining edges have a convective boundary condition, with ambient temperature $u_\infty = 293\,\mathrm{K}$. The right edge has a heat transfer coefficient $h_1 = 100\,\mathrm{W/m^2K}$ whereas bottom and top edges have $h_2 = 10\,\mathrm{W/m^2K}$.

The problem is then solved with a mesh that does not conform to the microvascular network (see Figure 3.17b) using $\{1, \xi, \eta\} \times \{1, r_1\}$ and $\{1, \xi, \eta\} \times \{r_i\}_{i=1}^3$ as the sets for interface and junction enrichments, respectively. The temperature distribution both considering the flow and neglecting it are illustrated in Figure 3.18 using the same scale. Injecting flow into the domain has the direct effect of reducing the temperature by about 25 K. Note that this formulation is able to capture the loss in symmetry with respect to the horizontal barycentric axis due to the fact that the fluid increases its temperature from the inlet to the outlet.

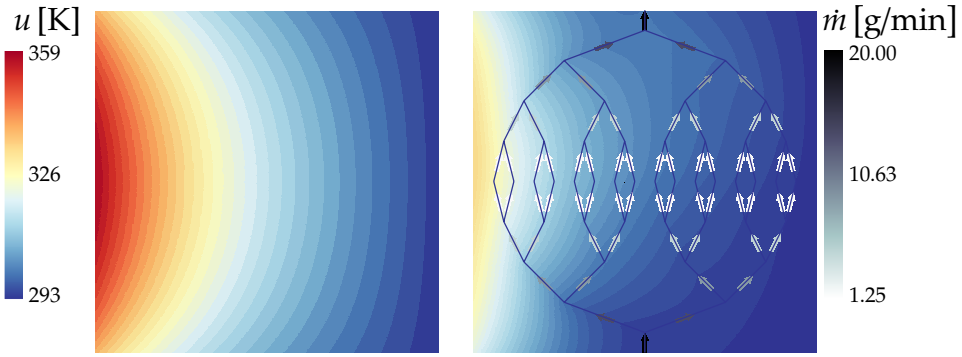


Figure 3.18 – Temperature distribution for a biomimetic active cooling material with flow (right) and without it (left). The problem was solved on a non-conforming mesh using $\{1, \xi, \eta\} \times \{1, r_1\}$ as the set for interface enrichments and $\{1, \xi, \eta\} \times \{1, r_i\}_{i=1}^3$ for junction enrichments.

3.5.2 Polycrystalline microstructure example

Even though all the problems studied so far have focused on line heat sources (or heat sinks), the enrichment functions presented in this work can also be used to study problems containing material interfaces. Consider the polycrystalline microstructure on a square domain Ω shown in Figure 3.19. A square domain is used again and the grains inside it have increasing conductivity values (in W/mK) $\kappa_1 = 2$, $\kappa_2 = 4$, $\kappa_3 = 8$ and $\kappa_4 = 380$. A uniform heat flux $\bar{q} = 100\,\mathrm{W/m}$ is applied over the top edge. The bottom edge is a convective boundary, with ambient temperature $u_\infty = 293\,\mathrm{K}$ and heat transfer coefficient $h = 100\,\mathrm{W/m^2K}$. The problem is solved using the enrichment functions $\{1, \xi, \eta\} \times \{1, r_i\}_{i=1}^{3}$, resulting in a quadratic approximation. The resulting temperature distribution is presented in Figure 3.20, clearly demonstrating the ability of the proposed GFEM model to capture the discontinuous temperature gradients along the grain boundaries, including those at triple junctions.

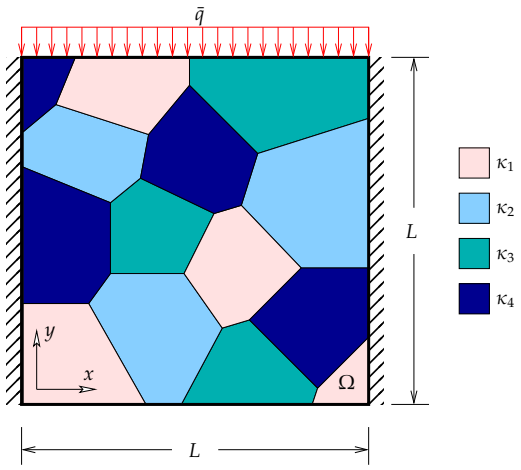


Figure 3.19 – Schematic for the polycrystalline microstructure example. The material is divided in grains having different thermal conductivity values. Boundary conditions include insulated left and right edges, a constant heat flux $\bar{q}$ on the top edge and a convective boundary along the bottom edge.

3.6 Conclusions

The enrichments introduced in this work solve the problem of having multiple interfaces converging to a single point inside finite elements. When there is only a single interface, these enrichments reduce to the ramp enrichments discussed in detail in Sections 3.4.1

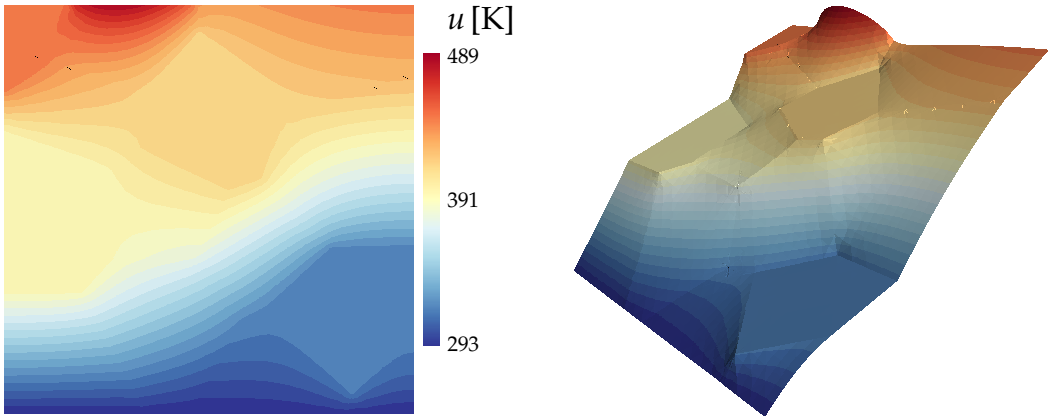


Figure 3.20 – Temperature distribution for the polycrystalline microstructure example emphasizing the discontinuous gradients along grain boundaries.

and 3.4.2. It was shown that the correction factor proposed by Fries in [76] is needed when ramp functions are used in a linear approximation. However, the ramp functions can only represent in this case a constant jump in the gradient of the field, so they fail to capture accurately the linear variation studied in Section 3.4.2 even when using the correction mentioned above. Also, the correction factor is not needed anymore when using a quadratic approximation and these functions recover optimal quadratic convergence rates.

It was shown that a single enrichment function (i.e., $R(\mathbf{x})$) for multiple interfaces does not recover optimal convergence rates. On the other hand, having one enrichment function per subdomain (i.e., $r_i(\mathbf{x})$) gives very accurate results. When the geometric representation of these subdomains is not available, junction enrichments can be built locally and used in conjunction with regular interface enrichments to provide more accurate results. It was shown in Section 3.4.3 that line bisectors have to be added because these enrichments are C^0 continuous along those lines as well. Adaptive integration can be used in the GFEM framework to find regions where the enrichment functions are not smooth. All results in this work reveal that quadratic approximations are more accurate than linear approximations for the same number of degrees of freedom.

Most examples studied in this work involved line heat sources in homogeneous materials to create the discontinuous gradient nature of the solution. The microvascular material example showed how this technique can be used in a real problem where the mesh is completely independent of the geometry of the network. The last example showed how the same enrichments can be used in heterogeneous materials. In this example the

discontinuity in the gradient results from having different conductivity values across the grains in a polycrystalline microstructure. Even though the presented work has focused entirely on the solution of the Poisson equation, the extension to elasticity problems is straightforward, i.e., the enrichment functions presented are general and they should work in the context of other physical phenomena. Although likely, the applicability of the enrichment functions presented in this work to 3D problems with line or planar heat sources remains to be demonstrated.

4 Multi-physics optimization

Multi-physics design of microvascular materials for active-cooling applications

Abstract

This paper describes a framework for the design of microvascular polymeric components for active-cooling applications. The design of the embedded networks involves complex and competing objectives that are associated with various physical processes. The optimization tool includes a PDE solver based on advanced finite element techniques coupled to a multi-objective constrained genetic algorithm. The resulting Pareto-optimal fronts are investigated in the optimization of these materials for void volume fraction, flow efficiency and maximum temperature objective functions.

4.1 Introduction

Numerous examples of vascular systems can be found in nature, e.g., in plants, animals and in the human body. Aside from transporting the nutrients and other necessary substances, the vascular system plays a fundamental role in thermoregulation [92], a fact widely studied in animals such as reptiles [93, 94]. In the human body, the heat produced by exercise is convected through the blood to the skin to be eliminated through evaporative cooling [95, 96]. Nature has served as a source of inspiration for the creation of structures and materials throughout history [1, 2, 6, 5]. Over the past decade, polymeric materials with embedded microvascular networks have been envisioned for mimicking the self-healing capability found in living organisms [10], and repeated healing on a polymer specimen using the aid of a three-dimensional microvascular network to supply the healing agent has been demonstrated [8]. These materials are currently being investigated for thermal regulation [97]. Due to the large surface to volume ratio of microchannels, microvascular cooling has been investigated for the cooling of integrated circuits and silicon computer chips [98, 99, 100], and micro-miniature refrigerators [101].

Microvascular polymeric materials are created by extruding a fugitive ink over a polymeric substrate, with tip diameters ranging from a few microns to hundreds of microns [10]. The resulting ink scaffold is then infiltrated with liquid polymer, until reaching the desired thickness of the final sample. Once the curing process of the polymer is finished, the ink is evacuated by heating (or cooling) the system, creating the void space that constitutes the network. The autonomic response then depends on the type of fluid used in the system. As mentioned earlier, by using a coolant as the fluid, microvascular materials can be used for active cooling applications [97]. The optimization of the topology of the microvascular network for such application encompasses a set of objectives that involve several physical phenomena. The energy that drives the flow through the network needs to be minimized in an optimal design. Moreover, for a specific configuration of thermal loads, the topology of the network needs to accommodate to the most efficient layout in order to keep the maximum temperature below a threshold value or to reduce the amount of thermal energy stored in the material. Finally, the topology also needs to be concerned with the volume of voids present in the final polymer, as this will have a direct impact on the stiffness and strength properties of the resulting system.

Several approaches have been used in the optimization of flow networks, and they can be grouped into three main categories: *i)* Semi-analytical approaches; *ii)* Gradient-based search methods; and *iii)* Evolutionary algorithms. Within semi-analytical methods, constructal theory [11, 12, 102] gives a practical alternative for the optimization of simple problems when there is information about optimal network configurations. As the complexity of the problem increases, the knowledge about optimal configurations decreases and so does the effectiveness of this approach. The use of constructal theory in the context of microvascular materials for active cooling can be found in [103, 104, 105]. The most common methodology within gradient-based search techniques, topology optimization, is applicable for problems for which the optimization process is concerned with continuous variables and the gradient computation is straightforward. As any gradient-based search technique, the algorithm usually converges to a local optimum. In order to find the global optimum, numerous attempts with different starting configurations are often needed, and even then the extremum is not guaranteed to be the global one. The use of topology optimization in the context of flow network design is discussed in [24, 106, 15]. Evolutionary algorithms are population-based algorithms, where a randomly generated set of candidate solutions evolves towards optimal structures throughout the optimization process. Genetic Algorithms (GAs), the most widely known methodology in this category, use *selection*, *crossover* and *mutation* as genetic operators. At each generation of the evolution process, the worst candidates are unlikely to be selected for reproduction, thereby

mimicking Darwin's *natural selection*. Selected individuals produce a new generation of candidate solutions by interchanging information, i.e., by applying the crossover operator. These new candidates then incur slight modifications in their structures through the mutation operator. Genetic algorithms do not have the drawbacks that the other methods have as they do not require any knowledge about optimal solutions, there is no need of a gradient computation, they search the entire decision space and they can be used for both continuous- and discrete-variable problems. Nevertheless, GAs are computationally demanding, often requiring millions of individual evaluations throughout their evolution. Fortunately, the evaluation step of the GA is *embarrassingly parallel*, which means that the candidate solutions can be evaluated independently in different processes, and thus GAs can obtain close to ideal speed-ups when running in distributed-memory computers. For a thorough discussion of GAs, we refer the reader to [16, 17].

A discussion on the use of GAs in the optimization of microvascular flow networks has been presented in Chapter 2, where a Multi-Objective Genetic Algorithm (MOGA) was chosen as the underlying optimization framework. The selection mechanism in a MOGA takes into account all objectives simultaneously and determines the best candidate solutions using the concept of *Pareto optimality*: when comparing two candidate solutions, the best candidate is better in at least one objective while not being worse in the other objectives. More specifically, the MOGA chosen in that work is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) introduced by Deb et al. [46]. A space of constraints was built into the original algorithm so that any number of constraints could be used in conjunction with the objective functions. In this sense, individuals are first selected according to their feasibility in the constraint function space, and if a decision cannot be made, the objective function space is analyzed next. The resulting algorithm provides a powerful tool for multi-objective constrained optimization, which can be used with an arbitrary number of objective functions and constraints. Moreover, analyzing a problem in two or three dimensions depends only on the microvascular network chosen as a template in the beginning of the optimization process. More details can be found in Chapter 2, reference [46], and the references therein.

For difficult problems, GAs usually involve thousands of individual evaluations to obtain good candidate solutions. As a result, the cost of the algorithm is directly proportional to the cost of evaluating the objective functions involved. By optimizing objective functions that describe different physical phenomena, this tool can be used to analyze truly multi-physics problems. This paper summarizes the authors' attempt to extend the results presented in Chapter 2 to include the thermal response of the embedded network so that optimized 2D structures can be obtained in the context of active cooling appli-

cations. The temperature field can be determined by solving the corresponding partial differential equations in both fluid and solid domains. In the literature, this problem is referred to as the *conjugate heat transfer problem* [99, 100]. To reduce the computational cost of the conjugate problem, we adopt in this work some simplifying assumptions that take advantage of the laminar regime of the flow and the high aspect ratio of the microchannels.

This paper is organized as follows: Section 4.2 discusses the mathematical representation of microvascular networks and states the network optimization problem. Section 4.3 describes the desired features for actively-cooled materials and introduces the objective functions and constraints considered in the optimization. Using a manufactured solution, the thermal solver is verified in Section 4.4. Section 4.5 gives a brief overview on the multi-objective constrained genetic algorithm used as the underlying optimization tool while various optimization examples are presented in Section 4.6.

4.2 Microvascular network representation

Within a polymeric component, there is an infinite number of possible network configurations, so the computational design requires considering only a finite number of them. We define as the optimization *network template* a microvascular network having all possible microchannel locations for the optimization being carried. This network is represented mathematically by a graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ represents the endpoint of one or more edges in the set $\mathcal{E}$, and each edge $e \in \mathcal{E}$ represents a microchannel of the network. Thus, for each $v \in \mathcal{V}$, there exists a physical coordinate $\mathbf{x}_v \subset \Omega \subset \mathbb{R}^2$, where Ω represents the matrix of the polymeric component. Other properties can be assigned to the vertices in the graph. For example, the pressure at a vertex $v \in \mathcal{V}$ is defined as a function $p_v : v(\mathcal{V}) \to \mathbb{R}$. Similarly, properties are associated with the edges of the graph, thus $D_e : e(\mathcal{E}) \to \mathbb{R}$ and $L_e : e(\mathcal{E}) \to \mathbb{R}$ denote the diameter and the length of an edge e, respectively. The distribution of the vertices within Ω can be done arbitrarily, but it is convenient to place them over a lattice layout $\Lambda := \left\{\sum_{i=1}^{2} z_i \mathbf{e}_i \mid z_i \in \mathbb{Z}\right\}, \Lambda \subset \Omega$, where $\mathbf{e}_i$ represent a basis of $\mathbb{R}^2$.

Consider the set of k discrete diameters $\mathcal{D} := \{D_i\}_{i=1}^{k}$, taken from the set of possible diameter tips used in the manufacturing process of the microvascular networks [10]. The zero diameter (i.e., the absence of the channel) can also be considered in the set $\mathcal{D}$, thus providing a means for the resulting networks to morph in topology. In other words, considering the zero-diameter case makes possible the evaluation of a finite number of subgraphs $\mathcal{G}' := (\mathcal{V}', \mathcal{E}')$ of the graph $\mathcal{G}$ that represents the network template, where

$\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. These definitions imply that, for a subgraph $\mathcal{G}'$, the *degree* $\deg(v')$ of a vertex representing the number of edges to which the vertex $v' \in \mathcal{V}'$ is connected, may not be the same as the degree for the corresponding vertex in the graph $\mathcal{G}$ of the network template (i.e., $\deg(v' \in \mathcal{V}') \leq \deg(v \in \mathcal{V})$). This fact has implications on the coverage of the network, as described in the following section. For now, let us borrow the following definitions from graph theory: the *minimum degree* of a graph $\mathcal{G}$, denoted $\delta(\mathcal{G})$, is defined as the minimum degree of its vertices (i.e., $\delta(\mathcal{G}) = \min_i \deg(v_i), \forall v_i \in \mathcal{V}(\mathcal{G})$); a graph $\mathcal{G}$ is *connected* if there exists a path for every pair of distinct vertices $v_i, v_j \in \mathcal{V}(\mathcal{G})$ through the existing edges of the graph $\mathcal{E}(\mathcal{G})$; finally, a *disconnected graph* $\mathcal{G}$ has at least two *connected components*, where each one of these components is a connected subgraph $\mathcal{G}'$ as defined earlier. The optimization process involves finding the diameter $D_i \in \mathcal{D}$ for all channels $e \in \mathcal{E}$ in order to optimize a set of objective functions $\Phi = \left\{\phi_i\right\}_{i=1}^{m}$, subject to a set of constraints $\Psi = \{\psi_i\}_{i=1}^{n}$. According to these definitions, there are $k^{|\mathcal{E}|}$ possible networks to be evaluated, where $|\mathcal{E}|$ is the number of edges in the set $\mathcal{E}$.

4.3 Design objectives and constraints

The design of microvascular polymeric materials is inherently an intricate process, for an ideal system must achieve several objectives. For example, it is desired that the microvascular network be *minimally invasive*, for the void of the network reduces the stiffness and strength properties of the resulting polymer. A desired feature might also be that the microvascular network distributes the fluid to all relevant parts of the matrix, thus providing *maximum coverage*. Furthermore, it might be desired that the topology of the final network design provides some degree of *redundancy*, so that the fluid can still reach all areas of the specimen even if damage or blockage occurs in certain parts of the network. It is also important that the energy needed to drive the flow through the microvascular network be minimized in an optimal design, thus *flow efficiency* is a required feature in these systems. Finally, in the context of active cooling applications, the structure of the network must be such that the *thermal energy* stored or the *maximum temperature* in the matrix are minimized for a given set of thermal loads. Intuitively, some of the design objectives discussed conflict with one another, so the optimization tool to be used needs to be able to optimize all objectives simultaneously, capturing the trade-offs between competing objectives. The following sections provide a mathematical representation of the objective functions of interest in the present study.

4.3.1 Flow efficiency

Microvascular networks are typically composed of microchannels with circular cross-sections and high aspect ratios (i.e., length to diameter ratio). Even in 3D scaffolds, aspect ratios of 6.25 have been reported [107]. Furthermore, due to the small diameter sizes and relatively small flow rates, the flow in these structures is laminar with a Reynolds number $Re < 1000$, a value which lies well below the onset of turbulent transition reported for these networks [108]. Figure 4.1 shows a schematic of the mathematical model used to represent a biomimetic microvascular material, defined over the domain $\Omega \subset \mathbb{R}^2$. The microvascular network template, denoted $\mathcal{G}$ on the figure, contains a single inflow vertex and a single outflow. More generally, let the network inlets be represented by the set of vertices $\mathcal{S} \subset \mathcal{V}(\mathcal{G})$ and the set of outlets by the vertex set $\mathcal{T} \subset \mathcal{V}(\mathcal{G})$. Then, $\dot{m}_{\mathcal{S}} := \sum_{v \in \mathcal{S}} \dot{m}_v$ and $\dot{m}_{\mathcal{T}} := \sum_{v \in \mathcal{T}} \dot{m}_v$ are the total mass flow rate entering and leaving the polymeric component, respectively, with $\dot{m}_{\mathcal{S}} = \dot{m}_{\mathcal{T}}$. An inset of the microchannel e shows the local coordinate system $\boldsymbol{\xi}_e = (\xi_e, \eta_e)$, and the flow velocity vector $\mathbf{v}_e(\boldsymbol{\xi}_e)$. Considering

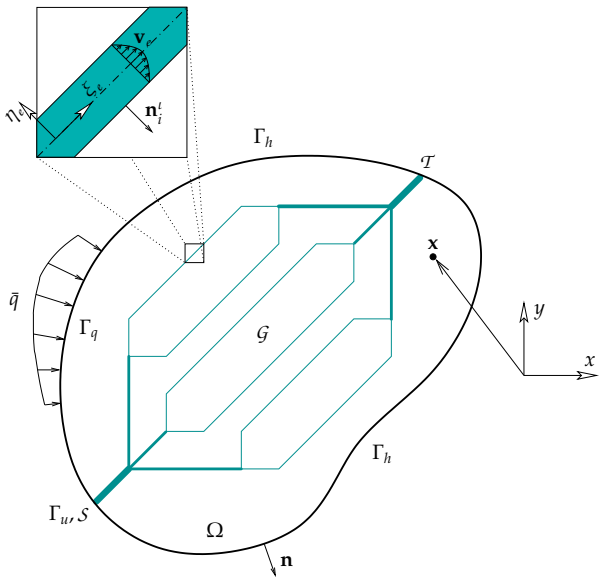


Figure 4.1 – Mathematical model of a biomimetic active cooling material. The material, defined over the domain $\Omega \subset \mathbb{R}^2$ with outward unit normal $\mathbf{n}$, has an embedded microvascular network, represented by a graph data structure $\mathcal{G}$. The domain contains an inflow $\mathcal{S}$ and an outflow $\mathcal{T}$. The total mass flow rate entering the material is $\dot{m}_{\mathcal{S}}$. An inset of microchannel e shows the velocity profile $\mathbf{v}_e(\boldsymbol{\xi}_e)$ assuming laminar flow. For the thermal boundary value problem, the boundary of the domain is split in mutually exclusive regions Γ_u, Γ_q and Γ_h, where Dirichlet, Neumann, and Robin boundary conditions are applied, respectively.

a fully developed flow of constant density ρ_f, the velocity vector in microchannel e of

diameter D_e and constant cross-section $A_e = \pi D_e^2/4$ is given by

$$\mathbf{v}_e(\xi_e) = 2\bar{v}_e\left(1 - 4\frac{\eta_e^2}{D_e^2}\right)\mathbf{e}_{\xi_e}, \tag{4.1}$$

where $\bar{v}_e$ is the mean velocity defined such that $\dot{m}_e \equiv \rho_f \bar{v}_e A_e$, with $\dot{m}_e$ denoting the mass flow rate in the channel. Then, the pressure drop between the end-vertices $v_j, v_k, \in \mathcal{V}$ of the channel is obtained using the Hagen-Poiseuille law,

$$\Delta p_e = \frac{128\nu L_e}{\pi D_e^4}\dot{m}_e, \tag{4.2}$$

where ν denotes the kinematic viscosity of the fluid. The assumption of fully developed flow is valid everywhere but close to inlet locations and microchannel junctions. The hydrodynamic entry length can be estimated [109] by

$$\xi_h \approx 0.05 D_e\, Re. \tag{4.3}$$

For the aspect ratios considered, the ratio $\xi_h/L_e = 0.005\, Re$ is very small so the flow can be considered as fully developed throughout the channel. Equation (4.2) can be rearranged to express the mass flow rate contribution at the end-vertices of the channel:

$$\begin{Bmatrix} \dot{m}_{v_j} \\ \dot{m}_{v_k} \end{Bmatrix} = \frac{\pi D_e^4}{128\nu L_e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} p_{v_j} \\ p_{v_k} \end{Bmatrix}. \tag{4.4}$$

More generally, the formulae above are also valid for the subgraphs of the microvascular network template, i.e., $\mathcal{G}' \subseteq \mathcal{G}$. Considering the contribution of $|\mathcal{E}'|$ microchannels, a system of linear equations $\vec{K}\vec{p} = \vec{\dot{m}}$ is assembled, where $\vec{K}$ is the network characteristic matrix of size $|\mathcal{V}'| \times |\mathcal{V}'|$, $\vec{p}$ the pressure vector and $\vec{\dot{m}}$ the mass flow rate vector [32]. Two types of boundary conditions are considered when solving this system of equations. The mass flow rate is prescribed over the vertex set $\mathcal{S} \subset \mathcal{V}'$, that represents network inlets, while network outlets, represented by the vertex set $\mathcal{T} \subset \mathcal{V}'$, have prescribed pressure values. Prescribed pressure values are typically set at zero, so that the computed pressure value at each network vertex corresponds to the difference with the outflow pressure. The pressure distribution in the network is fully determined after solving this system of linear equations.

Let us define the flow efficiency objective function $\phi_f : \mathcal{V}'(\mathcal{G}') \rightarrow \mathbb{R}$ as

$$\phi_f := \max_{v'_i, v'_j \in \mathcal{V}'} |p_{v'_i} - p_{v'_j}|. \tag{4.5}$$

In other words, the maximum pressure difference in the microvascular network represented by the subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ can be used to quantitatively describe the amount of energy needed to drive the flow through the network. The flow in each channel can also be computed once the pressure distribution is obtained by using Equation (4.2).

Equation (4.5) can be applied in two different contexts. The first one considers a constant value of the total mass inflow rate $\dot{m}_{\mathcal{S}}$. This case is straightforward and the objective function is obtained directly after solving the system of linear equations mentioned above. The second context considers a constant power

$$P := \frac{1}{\rho_f} \sum_{v \in \mathcal{S}} \dot{m}_v p_v, \tag{4.6}$$

where p_v represents the pressure at inlet locations (i.e., $v \in \mathcal{S}$) in excess of the value prescribed at outlet locations. In this case, the prescribed mass flow rates are scaled such that the power P remains constant for each network. In other words, instead of considering a constant mass inflow throughout the optimization, the power is fixed to the value of the template configuration P^0. The power P is computed for each individual after solving for the flow efficiency objective function using the initial flow boundary conditions, and a scaling factor is obtained as the ratio between the reference power P^0 and P. This factor is used to scale both the pressure and the mass flow rate, such that the power of the candidate solution equals that of the template configuration P^0.

4.3.2 Void volume fraction

As described previously, the microvascular network may have a detrimental impact on the stiffness and strength properties of the resulting polymer. To achieve a minimally invasive network, we aim to minimize the network void volume fraction objective function $\phi_v : \mathcal{E}'(\mathcal{G}') \rightarrow \mathbb{R}$, defined as

$$\phi_v := \frac{\sum_{e' \in \mathcal{E}'} L_{e'} D_{e'}}{A_\Omega}, \tag{4.7}$$

i.e., as the projected area of microchannels normalized by the area of the domain A_Ω. During the optimization process, groups of channels with no pressure differential (and therefore no flow) are eliminated prior to the computation of ϕ_v.

4.3.3 Thermal field

For the definition of the thermal boundary value problem, we refer again to Figure 4.1. The domain that represents the material $\overline{\Omega} \subset \mathbb{R}^2$ is defined such that $\overline{\Omega} := \overline{\Omega_s \cup \Omega_f}$, where Ω_s and Ω_f represent mutually exclusive parts of the domain occupied by the solid and the fluid, respectively. The boundary of the domain $\Gamma := \overline{\Omega} - \Omega$, with outward unit normal $\mathbf{n}$, is partitioned into regions Γ_u, Γ_q and Γ_h, where Dirichlet, Neumann and Robin boundary conditions are applied, respectively. The boundary regions are also mutually exclusive such that $\Gamma = \overline{\Gamma_u \cup \Gamma_q \cup \Gamma_h}$ and $\Gamma_u \cap \Gamma_q \cap \Gamma_h = \emptyset$. Finally, there are $|\mathcal{E}'|$ interfaces between the fluid and the solid domains, defined as $\mathcal{I} = \{\iota_i\}_{i=1}^{|\mathcal{E}'|} := \overline{\Omega_s \cap \Omega_f} \setminus \Gamma$, each one with normal vector $\mathbf{n}_i^\iota$.

The temperature distribution in the mathematical representation of the microvascular polymeric material can then be obtained by solving the conjugate heat transfer boundary value problem. This implies solving for the temperature in both solid and fluid domains, subject to a set of boundary conditions and ensuring continuity of the temperature and heat flux along solid-fluid interfaces. This requires a discretization in both domains and the exchange of information between them if a non-monolithic solver is used. To reduce the problem size, we adopt the following expression of the heat flow rate per unit length q_e at channel e, which can be obtained through an energy balance [90]:

$$q_e = \dot{m}_e c_p \frac{du}{d\xi_e}, \tag{4.8}$$

with c_p denoting the specific heat of the fluid, and $du/d\xi_e$ representing the gradient of the temperature in the direction of the channel. This equation assumes a fully developed temperature profile and also constant heat rate along the wall of the channel. The small diameters (of the order of tens to hundreds of microns) and the high aspect ratios of these microchannels allow us to mathematically treat them as line segments. Then, the simplified equation for the heat flow rate given by Equation (4.8) allows us to consider the cooling effect of the microchannels as heat sources (or sinks) over these line segments, without the need to solve explicitly for the temperature in the fluid. The strong form of the simplified thermal problem becomes: Given the density ρ_f and specific heat c_p of the fluid, the ambient fluid temperature u_∞, the heat transfer coefficient $h : \Gamma_h \to \mathbb{R}$, the thermal conductivity tensor in the solid $\boldsymbol{\kappa} : \overline{\Omega} \to \mathbb{R}^2 \times \mathbb{R}^2$, the heat source $f : \Omega \to \mathbb{R}$, prescribed temperature $\bar{u} : \Gamma_u \to \mathbb{R}$ and prescribed heat flux $\bar{q} : \Gamma_q \to \mathbb{R}$, find the temperature field in

the solid u such that

$$\nabla \cdot (\boldsymbol{\kappa} \nabla u) + f = \delta (\mathbf{x} - \mathbf{x}_e) \, c_p \dot{\mathbf{m}}_e \cdot \nabla u \qquad \text{on } \Omega, \tag{4.9}$$

with $\dot{\mathbf{m}}_e = \dot{m}_e \mathbf{e}_{\xi_e}$, and boundary conditions

$$u = \bar{u} \qquad \text{on } \Gamma_u, \tag{4.10}$$

$$\boldsymbol{\kappa} \nabla u \cdot \mathbf{n} = \bar{q} \qquad \text{on } \Gamma_q, \tag{4.11}$$

$$\boldsymbol{\kappa} \nabla u \cdot \mathbf{n} = h (u_\infty - u) \qquad \text{on } \Gamma_h, \tag{4.12}$$

where the Dirac delta function is introduced to represent the location of the microchannels in the 2D domain. The boundary value problem given by Equations (4.9) through (4.12) is solved using a finite element formulation, as described in Section 4.4.

The maximum temperature objective function $\phi_T : \Omega^h \to \mathbb{R}$ is defined as

$$\phi_T := \max u^h (\mathbf{x}) , \qquad \forall \mathbf{x} \in \Omega^h, \tag{4.13}$$

where Ω^h is a finite element discretization of Ω and u^h is an approximate solution of u obtained from a discrete formulation.

In most applications, the source term f in Equation (4.9) is zero. However, there is a class of problems for which the 2D domain represents a microvascular fin surrounded by a fluid at temperature u_∞. In that case, the source term takes the form $f = h_f (u_\infty - u)$, where h_f denotes the film coefficient describing the heat transfer between the fin and its surrounding. For this class of design problems, we introduce the surface convected energy objective function $\phi_s : \Omega^h \to \mathbb{R}$ as

$$\phi_s := \frac{1}{\Omega^h} \sum_{\alpha=1}^{M} \int_{\Omega_\alpha} h_f \left(u^h (\mathbf{x}) - u_\infty \right) d\Omega. \tag{4.14}$$

where M denotes the number of elements that discretize Ω^h.

4.3.4 Connectivity constraint

The objective functions introduced above are computed only for connected graphs. When the candidate solutions are allowed to morph in structure throughout the optimization process, many of the subgraphs $\mathcal{G}' := (\mathcal{V}', \mathcal{E}')$, of the graph $\mathcal{G}$ that represents the network template become unfeasible because $\mathcal{G}'$ is not a connected graph. In this case the candidate

solution is marked as unfeasible. A very simple constraint can be obtained as

$$\psi_c := \begin{cases} 1 & \text{if } \mathcal{G}' \text{ is disconnected,} \\ 0 & \text{otherwise.} \end{cases}$$

However, this simple constraint is modified in practice so that it is easier to move away from the unfeasibility region. A better constraints relies on the number of connected components of the subgraph:

$$\psi_c := c\left(\mathcal{G}'\right) - 1,$$

where the function $c : \mathcal{G}' \to \mathbb{Z}$ gives the number of connected components. Finally, the number of edges in the subgraph can be used to move away from the unfeasibility region, considering the fact that the template network, which contains the maximum possible number of edges, is feasible:

$$\psi_c := \begin{cases} 1/\left|\mathcal{E}'\left(\mathcal{G}'\right)\right| & \text{if } \mathcal{G}' \text{ is disconnected,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

Equation (4.15) is used to impose the connectivity constraint in all optimization examples studied in this work.

4.4 Thermal solver

For the weak formulation, let $\mathcal{U} = \{u \mid u|_{\Gamma_u} = \bar{u}\} \subset H^1(\overline{\Omega})$ be the set of trial solutions for the temperature field and $\mathcal{V} = \{v \mid v|_{\Gamma_u} = 0\} \subset H^1_0(\overline{\Omega})$ be the variation space. The weak form of the problem is: Given ρ_f, c_p, u_∞, h, $\boldsymbol{\kappa}$, f, $\bar{q}$, $\bar{u}$ as defined earlier in Section 4.3.3, find $u \in \mathcal{U}$ such that

$$a\left(w,u\right) + a\left(w,u\right)_{\Gamma_h} = \left(w,f\right) + \left(w,\bar{q}\right)_{\Gamma_q} + \left(w,u_\infty\right)_{\Gamma_h} \qquad \forall w \in \mathcal{V}, \tag{4.16}$$

where the bilinear and linear forms are given by

$$a(w,u) = \int_{\Omega} \left(\nabla w \cdot (\boldsymbol{\kappa} \nabla u) + w\, \delta(\mathbf{x} - \mathbf{x}_e)\, c_p \dot{\mathbf{m}}_e \cdot \nabla u \right) d\Omega, \tag{4.17}$$

$$a(w,u)_{\Gamma_h} = \int_{\Gamma_h} whu \, d\Gamma, \tag{4.18}$$

$$(w,f) = \int_{\Omega} wf \, d\Omega, \tag{4.19}$$

$$(w,\bar{q})_{\Gamma_q} = \int_{\Gamma_q} w\bar{q} \, d\Gamma, \tag{4.20}$$

$$(w,u_\infty)_{\Gamma_h} = \int_{\Gamma_h} whu_\infty \, d\Gamma. \tag{4.21}$$

Equation (4.16) is discretized using a finite element formulation. It is worth noting that the resulting matrix in the discrete form is not symmetric due to the convective term in the bilinear form of Equation (4.17).

Let us partition the domain in M finite elements with a total of N nodes such that $\Omega^h \equiv \text{int}\left(\cup_{\alpha=1}^{M} \overline{\Omega}_\alpha\right) \approx \Omega$. A more advanced framework can be used to obtain the temperature field. The Generalized Finite Element Method (GFEM) enables us to obtain the temperature field in finite element meshes that are completely independent of the geometry of the microvascular networks. Because the topology of the optimal network is not known a priori, it is desirable that the finite element mesh used to solve the thermal field be completely independent of the location of the microchannels. Enrichment functions used in the GFEM that recover optimal convergence rates, developed especially for this work, have been proposed in Chapter 3. In the GFEM framework, the temperature field approximation becomes

$$u^h(\mathbf{x}) = \sum_{\alpha=1}^{N} \varphi_\alpha(\mathbf{x})\, \tilde{U}_\alpha + \sum_{\alpha=1}^{N} \varphi_\alpha(\mathbf{x}) \sum_{i=1}^{E} L_{\alpha i}(\mathbf{x})\, \hat{U}_{\alpha i}, \tag{4.22}$$

where the first term corresponds to the standard finite element interpolation (with standard degrees of freedom $\tilde{U}_\alpha$), the second term to the enriched/extended part of the approximation (with enrichment degrees of freedom $\hat{U}_{\alpha i}$), and the functions $\{L_{\alpha i}(\mathbf{x}) : \omega_\alpha \to \mathbb{R}\}_{i=1}^{E}$, with $\omega_\alpha := \{\mathbf{x} \mid \varphi_\alpha(\mathbf{x}) \neq 0\}$ being the *cloud* or *support* of $\mathbf{x}_\alpha$, attempt at capturing some *a priori* information about the solution of a problem. See also [57] for more details on the GFEM.

For the discussion that follows, polynomial and ramp enrichment functions (see Chap-

ter 3) are used for the GFEM:

$$\mathcal{L} = \{1, \xi, \eta\} \times \{1, r_i\}_{i=1}^{2}, \tag{4.23}$$
$$\xi = \frac{x - x_\alpha}{h_\alpha}, \quad \eta = \frac{y - y_\alpha}{h_\alpha}$$

where (x_α, y_α) are the coordinates of the node $\mathbf{x}_\alpha$, h_α is a scaling parameter that is proportional to the size of the cloud or support of node $\mathbf{x}_\alpha$, and r_i are the ramp functions described in Chapter 3. This choice of enrichments results in an approximation that recovers optimal quadratic convergence.

A manufactured solution is constructed to verify the implementation of the weak formulation given by Equation (4.16). Consider the domain $\Omega := L_x \times L_y$ shown in Figure 4.2a. Assume that along the line $y = \alpha L_y$, with $0 < \alpha < 1$, heat is being generated in proportion to the $x-$derivative (i.e., $f_1 = C\, \partial u/\partial x$, with C constant), following the reasoning behind Equation (4.8). Thus, along that line, the jump in the heat flux equals the generated heat:

$$\kappa \left[\!\left[\frac{\partial u}{\partial y} \right]\!\right]_{y=\alpha L_y} = C \left. \frac{\partial u}{\partial x} \right|_{y=\alpha L_y}.$$

It can be shown that a temperature satisfying the equation above can be written as

$$u(x, y) = e^{-\kappa x / C \alpha L_y (1-\alpha)} \left[\frac{y}{\alpha L_y} - \frac{y - \alpha L_y}{\alpha (1-\alpha) L_y} \mathcal{H}\left(y - \alpha L_y\right) \right], \tag{4.24}$$

where $\mathcal{H}(\cdot)$ is the Heaviside step function. Substituting Equation (4.24) into the heat equation gives the form of the heat source f needed to satisfy the equation.

The temperature field of Equation (4.24) is illustrated in Figure 4.2b for $\alpha = 0.5$. Approximate solutions are also given in Figures 4.2b and 4.2c for a matching and a non-matching mesh, respectively. In the matching case, a finite element mesh of 3-node triangles is used, where the edges of the elements follow the heat source applied along the line. For the non-matching case, the generalized finite element method is used, with a quadratic approximation given by the use of the enrichment functions in Equation (4.23). A mesh of 3-node triangles is used as the underlying mesh, hence with a linear *partition of unity*.

4.5 Genetic algorithms

With the formulation of the objective functions and constraints at hand, we now turn to the underlying optimization methodology. Genetic algorithms have been chosen for

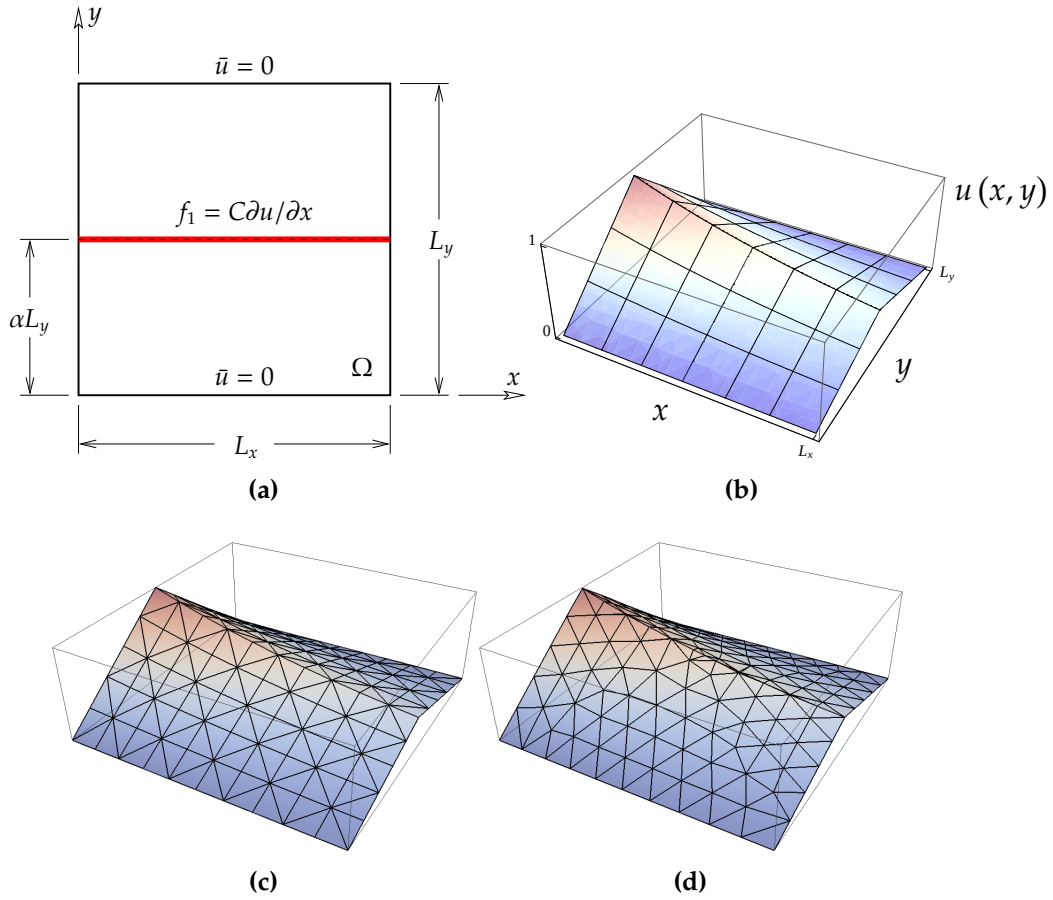


Figure 4.2 – Manufactured solution. (a) Schematic of the domain used, showing a heat source f_1 proportional to the $x-$derivative. (b) Exact temperature distribution given by Equation (4.24). (c) Finite element approximation on a matching mesh of 3-node triangular elements. (d) Generalized finite element quadratic approximation on a non-matching mesh of 3-node triangular elements using $\{1, \xi, \eta\} \times \{1, r_i\}_{i=1}^{2}$ as enrichment functions.

this work for a number of reasons. First, GAs do not require any a priori knowledge about optimal solutions. However, such knowledge can be exploited by creating a better starting guess for the population of individuals. The encoding of the problem gives powerful flexibility so the method can be used to solve both continuous- and discrete-variable problems. As mentioned earlier, GAs can be combined with a Pareto-selection mechanism to create Multi-Objective Genetic Algorithms (MOGAs). In these algorithms, individuals evolve all objectives simultaneously and the end result is usually a Pareto-optimal front of individuals, where all candidates are optimal solutions to the multi-objective optimization problem. The addition of a space of constraints creates a powerful tool for constrained multi-objective optimization, that can be used with an arbitrary number of objectives and constraints. GAs search the entire decision space so the optimal solutions found are likely to be global optima. Finally, the algorithm can be combined with other optimization procedures (e.g., gradient-based search techniques such as topology optimization or even another type of evolutionary algorithm) resulting in a *hybrid* methodology. In this regard the GA is used to search globally while the other search technique is used to find the local optimum. GAs have the disadvantage of consuming vast computational resources. However, GAs are embarrassingly parallel so a considerable speed-up can be obtained when running the algorithm in distributed-memory systems. The parallelization of GAs is challenging for problems where the evaluation time (i.e., the computational time required to evaluate objective functions and constraints) varies among individuals, requiring advanced load balancing techniques among processors. The parallelization scheme adopted in this work is discussed later in this section.

Genetic algorithms evolve a population $\mathcal{P}$ of candidate solutions, or *individuals* $\mathcal{I}$ in GA terminology. Each individual contains a *chromosome* C, that encodes in each one of its *alleles* $\mathcal{A}$ the candidate solution. The encoding used in this study is discussed next. As explained in Section 4.2, the optimization starts with a microvascular network template. This network is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with a total of $|\mathcal{E}|$ possible microchannels. Each chromosome that represents a candidate solution has length equal to the number of microchannels (i.e., $|C| = |\mathcal{E}|$), and each of its alleles gets a value from a diameter set $\mathcal{D} = \{D_i\}_{i=1}^{k}$ that contains the possible diameter choices. In GA terminology, this is termed *k−ary encoding*. As explained previously, considering the zero diameter allows the network to morph in structure from the network template. The mapping resulting from applying a chromosome C to the graph $\mathcal{G}$ can be thought of as a function $f : \mathcal{G} \rightarrow \mathcal{G}' \subseteq \mathcal{G}$, resulting in a subgraph with fewer edges when considering the zero diameter.

The adopted multi-objective GA, outlined in Algorithm 2, is called the Non-dominated

Sorting Genetic Algorithm II (or NSGA-II) and is attributed to Deb et al. [46]. Given the

Algorithm 2 Non-dominated Sorting Genetic Algorithm

procedure EVOLVE($|\mathcal{P}|, t_{\max}$)
  $\mathcal{P}_0$ = RANDOM($|\mathcal{P}|$)
  Q_0 =APPLYGENETICOPERATORS($\mathcal{P}_0$)
  while $t < t_{\max}$ **do**
    $R_t = \mathcal{P}_t \cup Q_t$
    $\mathcal{F}_i$ = FASTNONDOMINATEDSORT(R_t)
    repeat
      CROWDINGDISTANCEASSIGNMENT($\mathcal{F}_i$)
      $\mathcal{P}_{t+1} = \mathcal{P}_{t+1} \cup F_i$
    until $|\mathcal{P}_{t+1}| < |\mathcal{P}|$
    sort ($\mathcal{P}_{t+1}, \lll$)
    $\mathcal{P}_{t+1} = \mathcal{P}_{t+1}[0 : |\mathcal{P}|]$
    Q_{t+1} =APPLYGENETICOPERATORS($\mathcal{P}_{t+1}$)
    $t \leftarrow t + 1$
  end while
end procedure

population size $|\mathcal{P}|$, the algorithm starts by creating a random population of individuals $\mathcal{P}_0$. A children population Q_0 is then created by applying genetic operators *selection*, *crossover* and *mutation*. Then the algorithm enters a loop over a maximum number of iterations (or *generations*) $t_{\max}$, where the initial population evolves over time towards better candidate solutions. At each generation, the combined parent-children population $\mathcal{R}_t$ is split into sets $\mathcal{F}_i$ (non-dominated fronts) according to their domination rank (see below). The population for the next generation is then obtained by considering individuals starting from the best fronts until the population size equals $|\mathcal{P}|$, considering also a *crowding distance* parameter, used to populate the resulting fronts better. The genetic operators are applied again to obtain the children population for the next generation and the process is repeated.

The original algorithm was modified in the present study in order to handle constraints by using the operator $\lll$ defined as follows: Given the objective function space Φ, an individual $\mathcal{I}_i$ that is better than (or dominates) another individual $\mathcal{I}_j$ in that space, is expressed as $\mathcal{I}_i \ll_\Phi \mathcal{I}_j$. For this to be true, the corresponding objective functions are compared and individual $\mathcal{I}_i$ is better in at least one of them while being no worse in the rest. This concept, due to Vilfredo Pareto [45], is called *dominance* and it is borrowed from the field of economics. Similarly, considering the constraint space Ψ, $\mathcal{I}_i \ll_\Psi \mathcal{I}_j$ implies that individual $\mathcal{I}_i$ dominates individual $\mathcal{I}_j$ in the constraint function space. At any generation $t < t_{\max}$, the population $\mathcal{P}$ can be split into two mutually exclusive sets (i.e., $\mathcal{P} = \mathcal{P}^F \cup \mathcal{P}^U$, $\mathcal{P}^F \cap \mathcal{P}^U = \emptyset$), where $\mathcal{P}^F$ and $\mathcal{P}^U$ contain only feasible and unfeasible

individuals, respectively. The operator $\ll$ considers both spaces, so the domination of individual $\mathcal{I}_i$ over individual $\mathcal{I}_j$ is expressed as follows:

$$\mathcal{I}_i \ll \mathcal{I}_j \quad \Leftrightarrow \quad \begin{cases} \mathcal{I}_i \in \mathcal{P}^F, & \mathcal{I}_j \in \mathcal{P}^U, \\ \mathcal{I}_i, \mathcal{I}_j \in \mathcal{P}^U, & \left(\mathcal{I}_i \ll_\Psi \mathcal{I}_j\right) \text{ or } \left(\mathcal{I}_i \not\ll_\Psi \mathcal{I}_j \text{ and } \mathcal{I}_i \ll_\Phi \mathcal{I}_j\right), \\ \mathcal{I}_i, \mathcal{I}_j \in \mathcal{P}^F, & \mathcal{I}_i \ll_\Phi \mathcal{I}_j. \end{cases} \tag{4.25}$$

This operator is also used during the selection process when the genetic operators are applied. Finally, the operator $\lll$ considers also the crowding distance assignment, so that

$$\mathcal{I}_i \lll \mathcal{I}_j \quad \Leftrightarrow \quad \left(\mathcal{I}_i \ll \mathcal{I}_j\right) \text{ or } \left(\mathcal{I}_i \not\ll \mathcal{I}_j \text{ and } c_i > c_j\right), \tag{4.26}$$

where c_i and c_j are the crowding distance values for individuals $\mathcal{I}_i$ and $\mathcal{I}_j$, respectively, as discussed in Chapter 2. In Equations (4.25) and (4.26), $(\cdot \not\ll \cdot)$ implies non-domination, e.g., $\mathcal{I}_i \not\ll_\Psi \mathcal{I}_j$ implies also $\mathcal{I}_j \not\ll_\Psi \mathcal{I}_i$.

Parallel implementation

As the size of the problem increases, the parallelization of the GA is required if a Pareto-optimal front is to be obtained in a reasonable amount of time. The following analysis follows the methodology presented by Cantú-Paz and Goldberg [49]. Let us assume a master-slave approach for the parallelization, where slave processes are in charge of the evaluation of all objective functions and constraints for the individuals they receive and the master process receives this information and applies genetic operators. Considering only the evaluation part of the GA, the total parallel computational time is

$$T_p = (p-1)\, t_c + \frac{\bar{t}_e |\mathcal{P}|}{p-1}, \tag{4.27}$$

where t_c is the communication time, $\bar{t}_e$ the average evaluation time, $|\mathcal{P}|$ is the population size and p the number of processes. The optimal number of processes to solve this problem is given by

$$p^\star = 1 + \sqrt{\gamma |\mathcal{P}|}, \tag{4.28}$$

with $\gamma = \bar{t}_e/t_c$. The speed-up $S_p = T_s/T_p$, defined as the ratio between sequential and parallel computational times, is then

$$S_p = \frac{\gamma\,|\mathcal{P}|}{(p-1) + \gamma\,|\mathcal{P}|/\,(p-1)}. \tag{4.29}$$

Note that the inherent part of the GA (i.e., the time spent on genetic operators and in the NSGA-II front determination) is neglected in this analysis, so Equation (4.29) reflects only the speed-up in the evaluation. For the problems investigated in this work, the time spent in the inherent part of the GA can indeed be neglected and Equation (4.29) gives a good estimation of the speed-up. Figure 4.3 shows the speed-up S_p and efficiency $E_p = S_p/p$ as a function of p for $\gamma = \{250x\}_{x=1}^{4}$ and a population size $|\mathcal{P}| = 1000$. This population size is typical for the type of optimization problems studied in this work. The figures show that using an arbitrary number of processes can have a detrimental impact on the total computational time due to communication time. The speed-up figure also shows the optimal number of processes given by Equation (4.28).

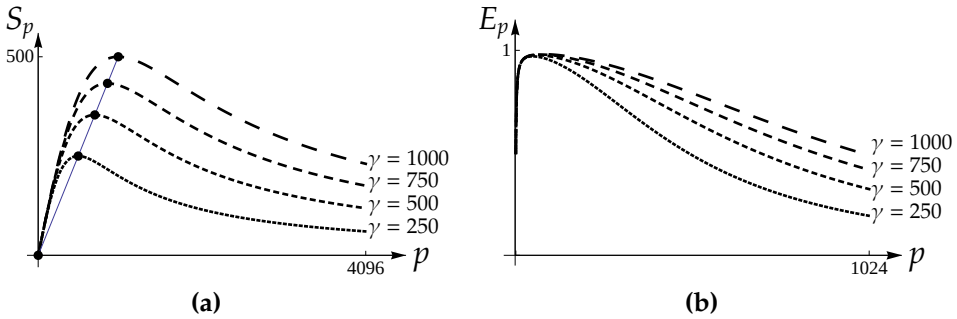


Figure 4.3 – (a) Speed-up $S_p = T_s/T_p$; and (b) Efficiency $E_p = S_p/p$, as a function of the number of processes p for a population size $|\mathcal{P}| = 1000$ and $\gamma = \{250x\}_{x=1}^{4}$.

It is worth mentioning that for problems where the individuals can have very different computation times, a naive implementation that considers no load balancing would reduce the efficiency, as defined above. In the present study, the evaluation times for the candidate solutions vary drastically, as all objective functions can only be computed for those individuals that are feasible. The time complexity of the algorithm used to find out the number of connected components in the network of the candidate solution is $O(\mathcal{V}' + \mathcal{E}')$, and this time is spent to find out if the individual is feasible or not. When a microvascular network that is not connected is found, the individual is marked as unfeasible and no time is spent computing the objective functions defined in Section 4.3. The objective functions studied in this work have very different time complexities. The

computation of the void volume fraction ϕ_v has complexity $O(\mathcal{E}')$. For the flow efficiency objective function ϕ_f, the time complexity is $O(\mathcal{V}'^3)$. Finally, the maximum temperature objective function ϕ_T and the surface convected energy objective function ϕ_s have complexity $O(N^3)$, with N being the number of nodes in the finite element mesh. Because the number of nodes in a mesh is clearly larger than the number of vertices in the network of a candidate solution, the evaluation of a feasible individual has complexity $O(N^3)$, which is much higher than that of an unfeasible individual $O(\mathcal{V}' + \mathcal{E}')$.

4.6 Results

As indicated earlier, a GA-based optimization of microvascular materials has been presented in Chapter 2, with emphasis on maximizing the flow efficiency and minimizing the void volume fraction of the embedded network. Here we extend that methodology to multi-physics optimization that includes the solution of the temperature field. Tournament selection, uniform crossover, and bit-wise mutation are used as genetic operators for the GA [16, 17]. In the following examples, water is used as the coolant, with density $\rho_f = 1000$ kg/m^3, kinematic viscosity $\nu = 1.05 \times 10^{-6}$ m^2/s and specific heat $c_p = 4185.5$ J/kg K.

4.6.1 Uniform heat trap example

This first application is used primarily as a verification problem and consists of a rectangular domain $\Omega = (26 \times 12)\,\text{mm}^2$, with two holes $\omega = (7 \times 2)\,\text{mm}^2$, as represented in Figure 4.4. The figure displays the microvascular network template, consisting of 446 channels. Flow boundary conditions include prescribed mass flow rate $\dot{m}_v = 10$ g/min $\forall v \in \mathcal{S}$ of water at three locations over the left edge and prescribed zero pressure at three locations over the right edge, as shown in the figure. For the thermal boundary value problem, a uniform heat source is applied to the domain strip between the two holes. All free edges in the domain have a convective boundary condition. Figure 4.4 also shows the temperature field in the absence of flow, where a maximum temperature $u_{\max} = 327$ K is reached. Thus, this problem tests the ability of the genetic algorithm to obtain a microvascular network configuration where the flow is collected into a single channel that passes through the central strip where the heat source is applied. Intuitively, this structure would minimize

the value of ϕ_T. The optimization problem is then stated as:

$$\begin{aligned} &\text{minimize} && \Phi = \left\{\phi_v, \phi_f, \phi_T\right\}, \\ &\text{such that} && \psi_c = 0. \end{aligned}$$

The diameter set used throughout the optimization is $\mathcal{D}\,[\mu\text{m}] = \{0, 350\}$, and the temperature of the fluid at inlet locations considered is $u_f = 283.15$ K. A population size $|\mathcal{P}| = 3200$ is chosen for the GA evolution. Results are reported for both constant mass inflow rate and constant power.

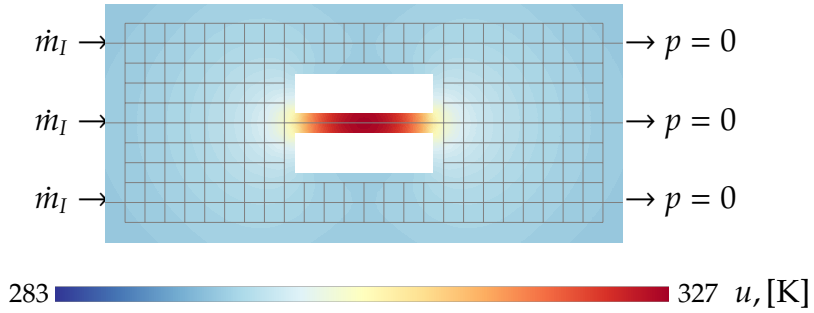


Figure 4.4 – Microvascular network template with 446 microchannels showing the flow boundary conditions and temperature solution in the absence of flow for the uniform heat trap example.

Constant inflow results

The evolution of the objective functions (normalized with those corresponding to the microvascular network template shown in Figure 4.4, ϕ_v^0, ϕ_f^0 and ϕ_T^0) is illustrated in Figure 4.5. The figure omits the first ten generations because they did not contain any feasible individuals (i.e., all candidate solutions contained disconnected networks). As apparent in the figure, the temperature objective function ϕ_T converges fast and remains roughly constant throughout the evolution. The other two objective functions need a higher number of generations to converge.

The entire Pareto-optimal front at generation $t = 2000$ is illustrated in Figure 4.6. The front contains two well defined branches, as illustrated clearly in the projection of the front on the $\phi_f - \phi_T$ plane. One of the branches remains roughly constant for the value of ϕ_T, and contains individuals for which the flow is entirely collected in microchannels

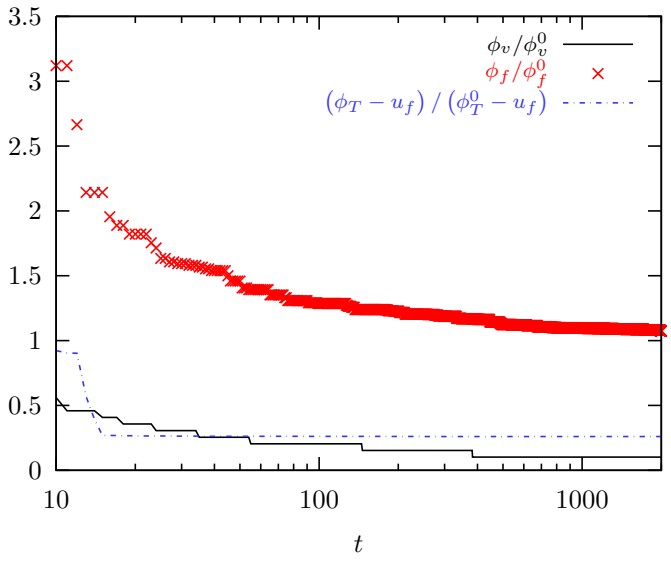


Figure 4.5 – Void volume fraction, flow efficiency and maximum temperature objective functions as a function of generations t for the uniform heat trap example. The objective function values are normalized with those of the microvascular template network ϕ_v^0, ϕ_f^0 and ϕ_T^0, and with the fluid temperature u_f.

passing through the central strip where the uniform heat source is applied. Thus, these candidate solutions are the best for minimizing the value of the maximum temperature ϕ_T. More precisely, individuals with increasing value of ϕ_v and decreasing value of ϕ_f have a slightly higher value of ϕ_T. Therefore, the individual that minimizes the maximum temperature objective function is not flow efficient. This individual, labeled (a) and marked with a □ symbol on the Pareto-optimal front, belongs to this first branch and it is revealed in Figure 4.7a. Arrows in Figure 4.7 represent the flow, so their sizes and gray scale are proportional to the flow magnitude. As expected, the flow is collected into a single microchannel. This individual not only minimizes the value of ϕ_T, but also the void volume fraction objective function ϕ_v. The other branch contains individuals where the flow is not collected entirely through the central strip, thus yielding a higher value of ϕ_T. However, individuals in this branch have a better value of the flow efficiency objective function ϕ_f because the flow is divided in multiple paths from inflow to outflow locations, which in turn increases the value of the void volume fraction ϕ_v. The individual shown in Figure 4.7b, which maximizes the flow efficiency at this particular generation, belongs to this branch and it is labeled (b) and marked with a ◯ symbol on the front. The structure of this individual is very close to that of the network template, thus having a very high value of the void volume fraction objective function ϕ_v. The ranges for the objective function values for this problem are as follows: ϕ_v [%] : $(2, 14)$, ϕ_f [MPa] : $(0.8, 5.7)$ and

ϕ_f [K] : [294.7, 313.1].

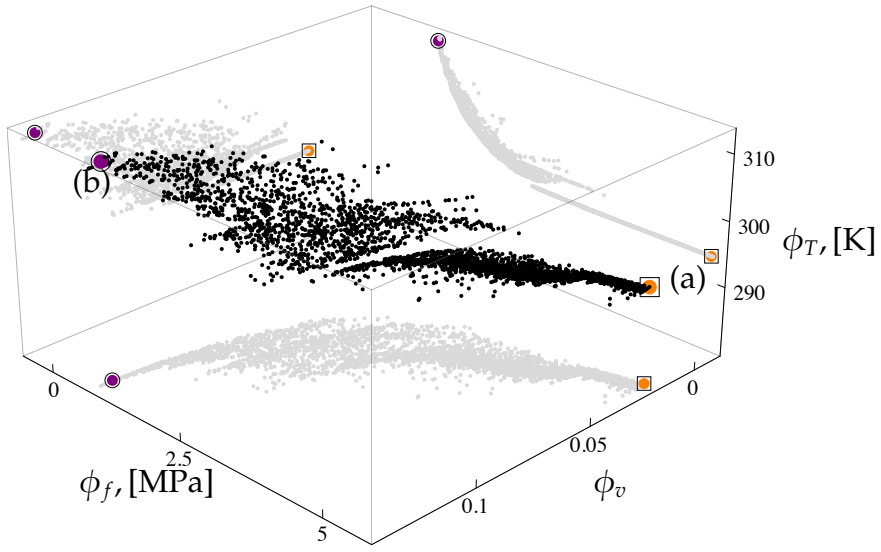


Figure 4.6 – Pareto-optimal front and its projection to three orthogonal planes after 2000 generations of the genetic algorithm for the uniform heat trap example under constant inflow conditions. Selected candidate solutions in the front correspond to those shown in Figures 4.7a and 4.7b, and are marked with symbols □, and ◯, respectively.

Constant power results

For the case of constant power, the entire front and its projections are presented in Figure 4.8 at generation $t = 4000$, and three of its candidate solutions (marked with symbols in the front and its projections) are illustrated in Figure 4.9. Note that the range of values for the flow in the latter is different from that shown in Figure 4.7. There are noticeable changes on the front, when comparing it to the front for constant inflow shown in Figure 4.6. The ranges for the objective functions are as follows: ϕ_v [%] : (2, 15), ϕ_f [MPa] : (0.75, 2.23) and ϕ_f [K] : [301.6, 326]. The pressure range changes drastically, reducing the upper bound considerably. However, this is just an apparent benefit because the temperature range is now considerably higher. The Pareto-optimal front is also divided in more than two branches, some of which contain just a few individuals. There is a branch that contains a single individual, the one that minimizes the void volume fraction (Figure 4.9a and individual marked with a ◇ in the front). This individual is very similar to that shown in Figure 4.7a for constant inflow, except for the fact that the flow is led through channels

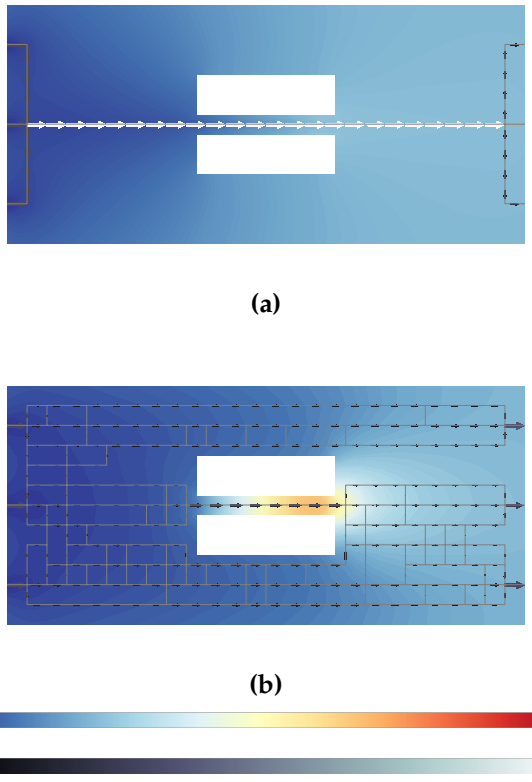


Figure 4.7 – Selected individuals at generation $t = 2000$ for the uniform heat trap with constant inflow: (a) candidate solution that minimizes both ϕ_v and ϕ_T; (b) individual that minimizes ϕ_f.

in the upper part of the domain and therefore has little effect on reducing the maximum temperature. With respect to the void volume fraction objective function, both networks are equivalent. Let us now focus on the tight branch with lower values of ϕ_T. Contrary to the constant inflow rate case, individuals with increasing value of ϕ_v and decreasing value of ϕ_f have a lower value of ϕ_T. Thus, the individual that minimizes ϕ_T is located at the opposite end of the branch, compared to the result for the constant inflow case. This individual, denoted by (b) on the front and marked with a □ symbol, is shown in Figure 4.9b. As for constant inflow, the flow is collected into a set of channels passing through the central strip where the heat source is applied. However, this time a dense network is created before and after the central strip to minimize the pressure drop in these regions. This individual contains the maximum value of the flow, which is roughly half of that for constant inflow. When considering constant power, the flow in the microchannels is reduced and therefore the maximum temperature increases. As a result, there is a difference of almost 7 K between this individual and the one displayed in Figure 4.7a. Yet another branch of the front contains individuals with high flow efficiency. The individual that minimizes the pressure drop, labeled (c) in the front and marked with a ◯ symbol, is shown in Figure 4.9c. This individual is very similar to the one shown in Figure 4.7b.

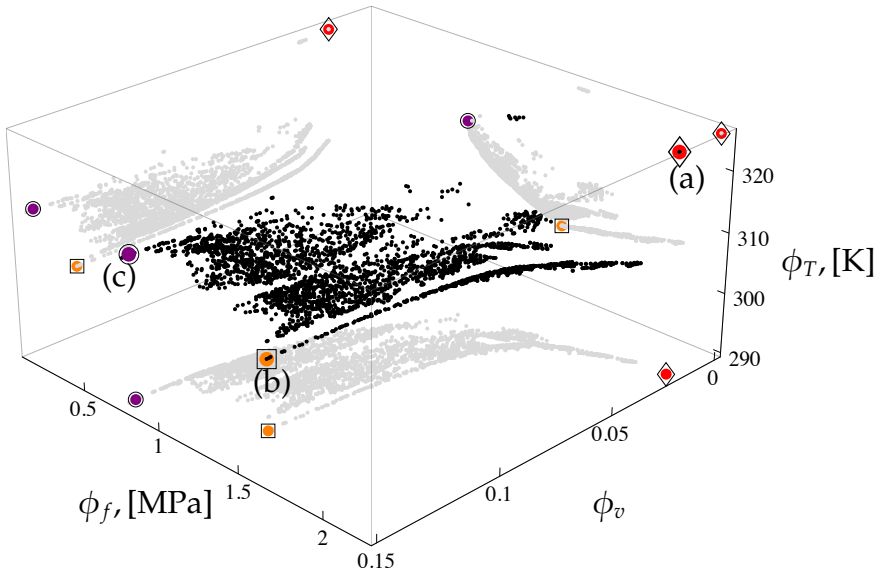
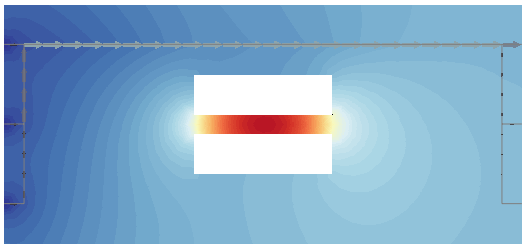
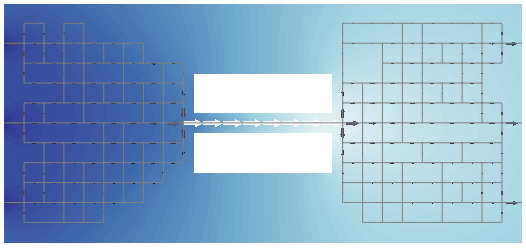


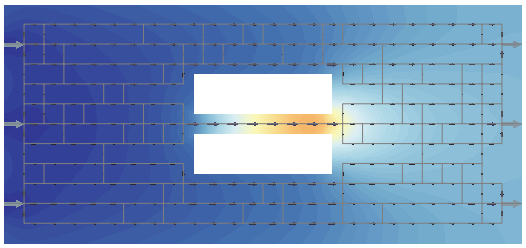
Figure 4.8 – Pareto-optimal front after 4000 generations of the genetic algorithm considering constant power in the uniform heat trap example. Selected candidate solutions in the front correspond to those shown in Figures 4.9a, 4.9b and 4.9c, marked with symbols ◇, □, and ◯, respectively.



(a)



(b)



(c)

Figure 4.9 – Selected individuals for the uniform heat trap example with constant power at generation $t = 4000$, minimizing: (a) ϕ_v; (b) ϕ_T; and (c) ϕ_f.

4.6.2 Localized heating example

Consider a rectangular domain $\Omega = (35 \times 20)\,\text{mm}^2$. For this 2D optimization problem, the microvascular network template contains 1241 microchannels (Figure 4.10a). There is a single inflow and a single outflow, located on the center of left and right edges, respectively. Flow boundary conditions include a prescribed mass inflow $\dot{m}_s = 10\,\text{g/min}$ of water and zero pressure at the outflow. For the thermal boundary value problem, all edges have a convective boundary condition and two localized heat sources are applied, as indicated in Figure 4.10a. Also, the inlet temperature of the fluid is set at $u_f = 283.15\,\text{K}$. The temperature distribution obtained using a finite element discretization in the absence of flow is also shown in Figure 4.10a, where a temperature difference of 59 K is observed. The diameter set considered in the optimization is $\mathcal{D}\,[\mu\text{m}] = \{0, 150\}$. As in the previous example, results are reported for constant mass inflow and for constant power. For the latter, the inflow is scaled such that the power of the candidate solutions equals that of the template configuration.

Constant inflow results

Figure 4.10 also reports candidate solutions obtained throughout the evolution of the genetic algorithm. Figures 4.10b, 4.10c and 4.10d represent the evolution of the candidate solution that minimizes the value of ϕ_T at generations $t = 100$, $t = 1000$ and $t = 10000$, respectively. It can be seen that the topology of the individual converges to a single path of interconnected microchannels going through the hot spots. Also, after reaching the first hot spot, the topology of the network is configured in such a way that the flow is taken over the cooler areas of the domain, allowing for the fluid to release some of its thermal energy. For the individual shown in Figure 4.10d, the maximum temperature is reduced by approximately 18 K. Due to its structure, it is expected that this solution has a poor performance for the flow efficiency objective function. Figures 4.10e and 4.10f also show optimal solutions at generation $t = 10000$. The individual presented in Figure 4.10e minimizes the void volume fraction, as it does its best to connect the inflow and outflow locations with a straight path of microchannels. The individual shown in Figure 4.10f maximizes the flow efficiency, as it divides the flow in as many channels as possible to the cost of increasing the void volume fraction.

Figure 4.11 presents the entire Pareto-optimal front for this problem at generation $t = 10000$. The front is again projected onto three orthogonal planes to assist with the visualization. The projection on the $\left(\phi_f, \phi_T\right)$ plane is very well defined, and there is a large

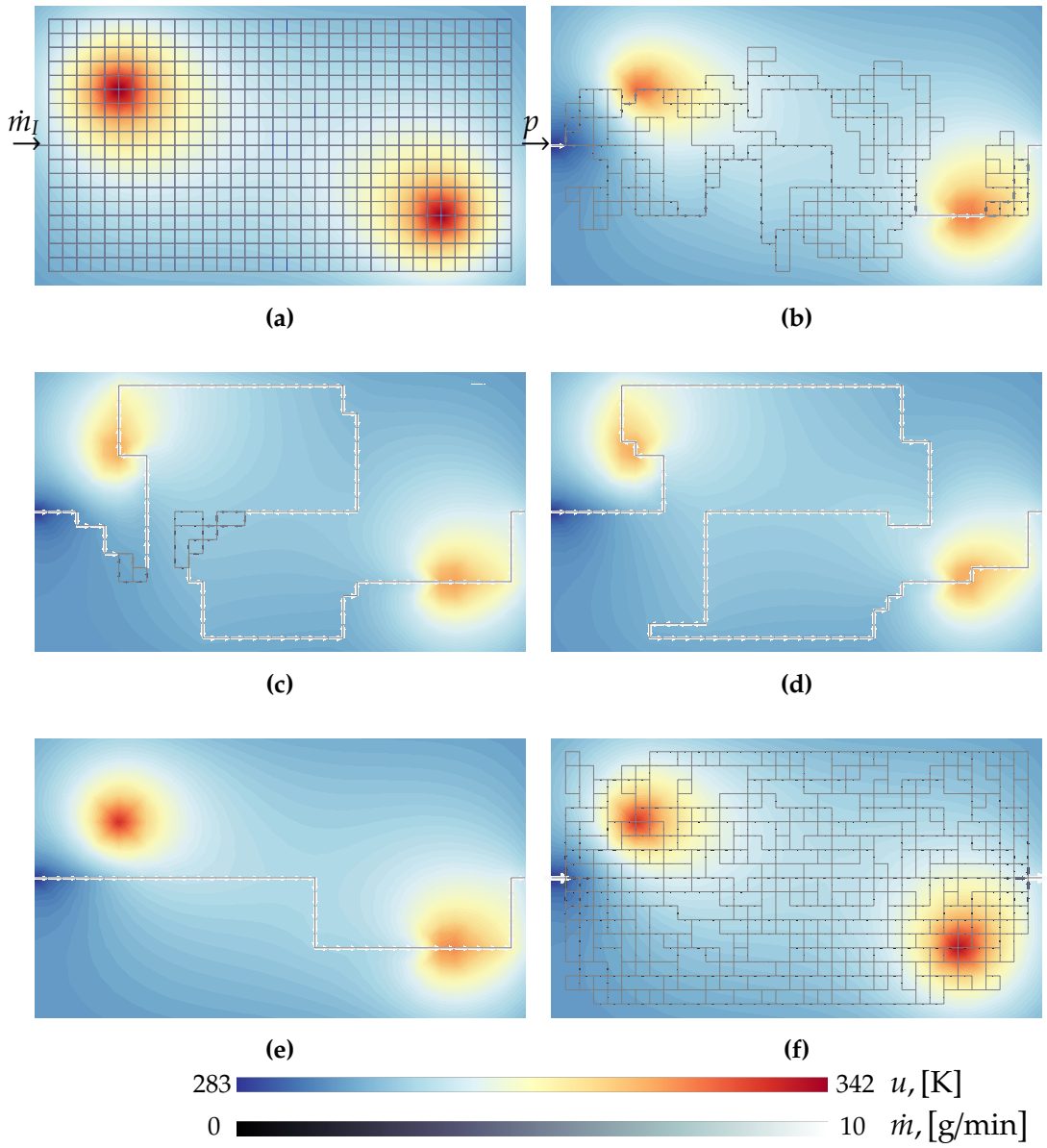


Figure 4.10 – Results for the localized heat source example for constant inflow. (a) Microvascular network template for this problem with 1241 channels and temperature solution without flow. Figures (b), (c) and (d) show the individuals that minimize ϕ_T at generations $t = 100$, $t = 1000$ and $t = 10000$, respectively. Figures (e) and (f) represent the solutions that minimize ϕ_v and ϕ_f at generation $t = 10000$, respectively. Arrows are sized proportionally to the flow magnitude.

range of individuals for which the maximum temperature objective ϕ_T remains almost constant. This compares similarly with the front obtained for constant inflow presented in Section 4.6.1. The same symbols as before are used, so Figures 4.10d, 4.10e and 4.10f are marked with □, ◇, and ◯ symbols, respectively. The objective function ranges for the front are: ϕ_v [%] : (0.96, 20], ϕ_f [MPa] : (0.08, 1.7) and ϕ_T [K] : [324.2, 342.2].

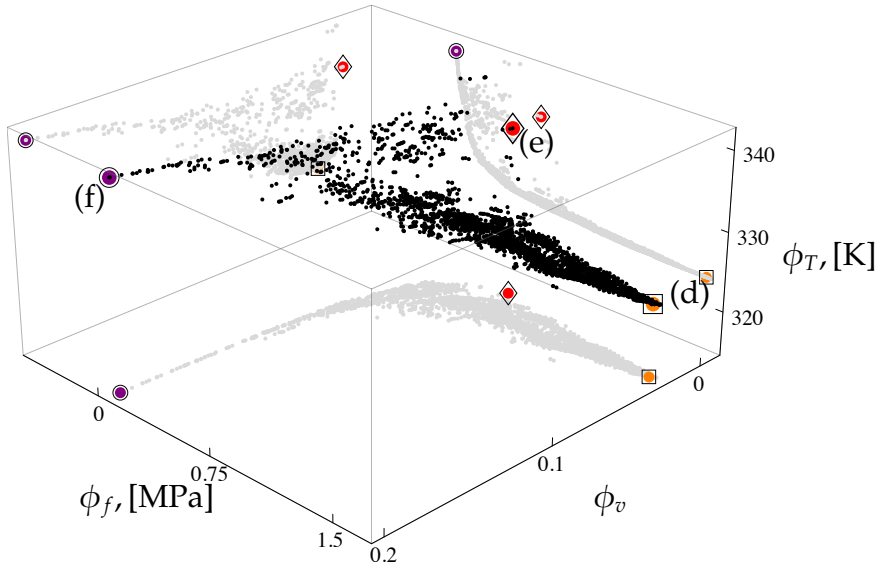


Figure 4.11 – Pareto-optimal front after 10000 generations of the NSGA-II for the localized heat example with constant inflow. Selected candidate solutions in the front, marked with □, ◇, and ◯ symbols, correspond to those shown in Figures 4.10d, 4.10e, and 4.10f, respectively.

Constant power results

Three candidate solutions at generation $t = 10000$ for the localized heat source considering constant power are illustrated in Figure 4.12. The genetic algorithm is able to capture the individual with the least possible void volume fraction in Figure 4.12a. The candidate solution presented in Figure 4.12b minimizes the value of ϕ_T. As in the case for constant inflow rate, the flow is directed to the hot spots, but this time with a dense network to reduce the amount of energy needed to drive the flow. It is worth mentioning that the entire flow is collected when reaching the second hot spot. However, since collecting the flow in a single path increases the pressure drop rapidly, the flow is only collected in a couple of microchannels. Finally, the individual outlined in Figure 4.12c minimizes the pressure drop, and it has a structure that again resembles that of the microvascular

network template.

The front in this case, shown in Figure 4.13, follows the same pattern as that of the previous example for constant power. It is clearly more segregated than the front shown in Figure 4.11. Once again, the individual that minimizes ϕ_T is located at the opposite end from that for constant inflow, for the set of individuals with low value of ϕ_T. This individual, labeled (b) on the front and marked with a □ symbol, corresponds to Figure 4.12b. It can be clearly seen in the $\phi_f - \phi_T$ plane that candidate solutions increase the value of ϕ_T as the pressure increases. Again, this is due to the fact that, as the network is less flow efficient, the flow decreases to keep the power constant, reducing also its ability to remove heat from the domain. The other two marked individuals on the front correspond to Figures 4.12a and 4.12c, and are marked with ◇ and ◯ symbols, respectively. The reported ranges for objective functions are: ϕ_v [%] : $(0.75, 22)$, ϕ_f [MPa] : $(0.072, 0.22)$ and ϕ_T [K] : $[335.9, 343.5]$. As it was the case for the previous example, the pressure range is drastically improved, to the cost of incrementing the temperature range. The upper bound on the pressure considering constant power is 13% of that for constant inflow, and there is a difference of almost 12 K between the lower bounds for the temperature.

A direct comparison between the thermal results presented in Figures 4.10d and 4.12b does not however account for the very large pressure penalty associated with the single channel solution obtained with the constant inflow case (Figure 4.10d). If indeed we scale down the driving pressure for that case to match the reference power value, the corresponding temperature drop would only amount to 0.3 K.

4.6.3 Microvascular fin

Consider the same domain studied in the previous problem, i.e., $\Omega = (35 \times 20)\,\text{mm}^2$, the same microvascular network template, and the same diameter set for the optimization. As in the previous example, flow boundary conditions include a single mass inflow of water $\dot{m}_s = 10\,\text{g/min}$ and zero pressure at a single outflow. This problem is studied in two separate cases, depending on how the flow boundary conditions are placed in the domain. The first scenario follows the previous example, placing the inflow and outflow on opposite sides of the domain. The second scenario places the flow boundary conditions on the same edge of the domain. Only constant inflow results are reported for both cases. For the thermal boundary value problem, the ambient temperature for this optimization problem is $u_\infty = 363.15$ K, thus the specimen is placed in a hot environment. As before, the inlet temperature of the fluid is set at $u_f = 283.15\,\text{K}$. For this optimization, energy is lost through the surface of the specimen only. Thus, convection through the surface

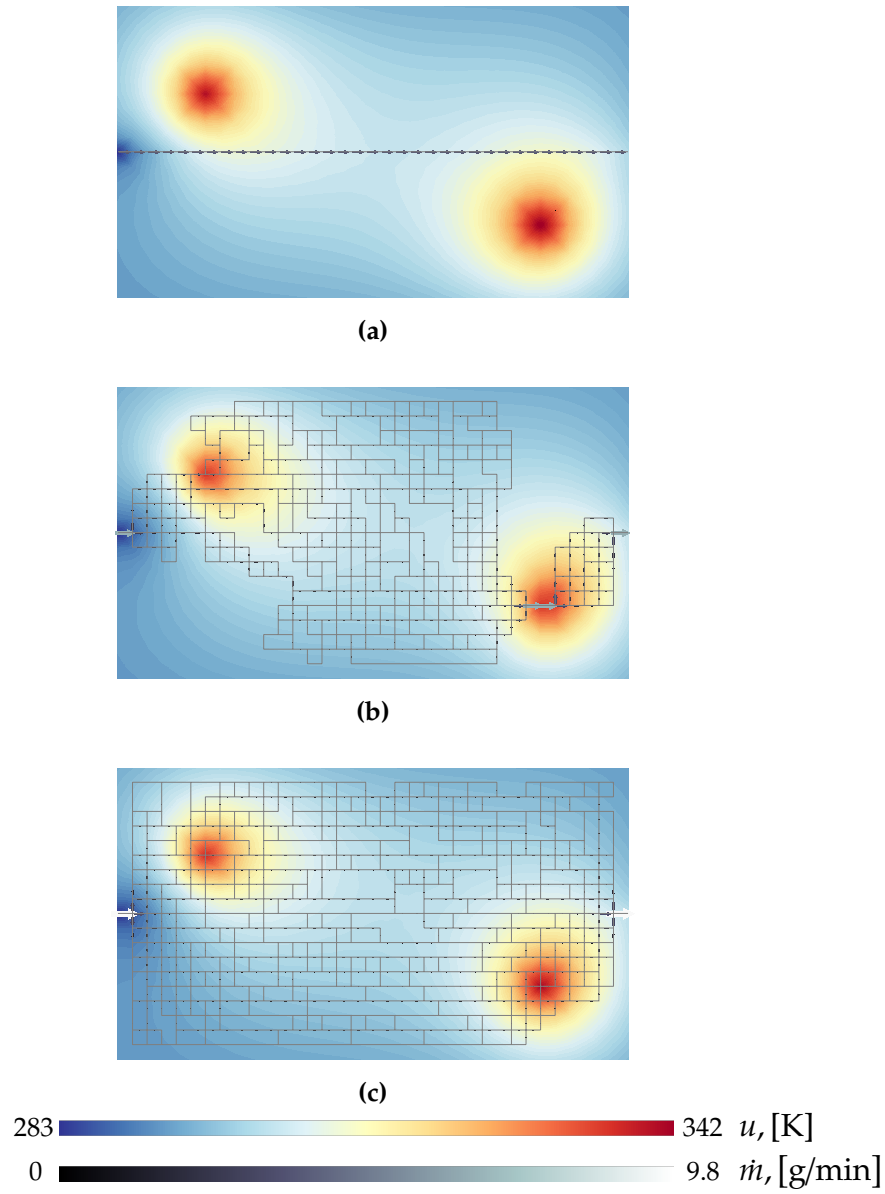


Figure 4.12 – Selected individuals at generation $t = 10000$ considering constant power for the localized heat source example. Labels (a), (b), and (c), respectively, denote optimal solutions with respect to the ϕ_v, ϕ_T, and ϕ_f objective functions.

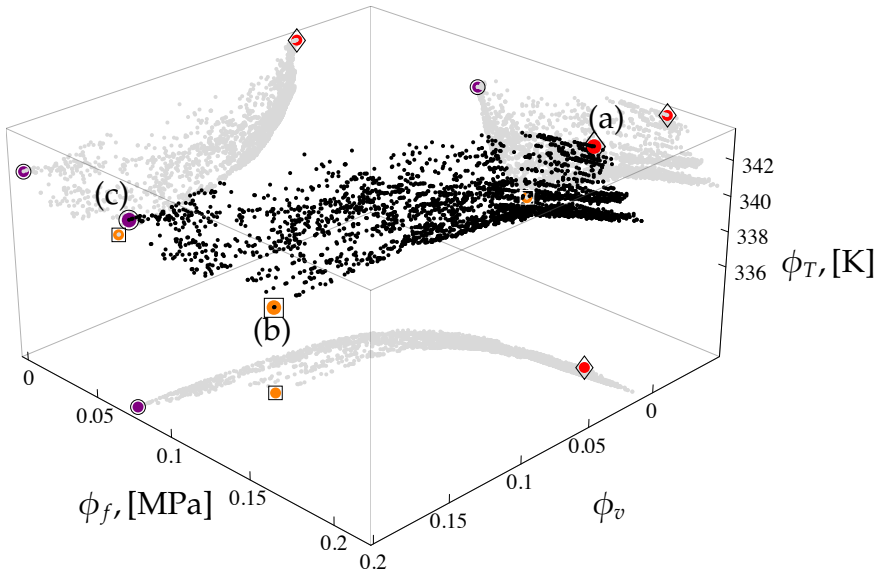


Figure 4.13 – Pareto-optimal front for the localized heat example with constant power at generation $t = 10000$. Selected candidate solutions, labeled (a), (b), and (c) in the optimal front, correspond to the individuals presented in Figures 4.12a, 4.12b and 4.12c. These solutions are also marked with ◇, □, and ◯ symbols, respectively.

is taken into account by having a temperature-dependent heat source $f = h_f\,(u_\infty - u)$, as explained in Section 4.3.3. The edges in this example are insulated, i.e., $\bar{q} = 0$. Thus, in the absence of flow, the temperature of the entire domain equals that of the environment. The optimization problem is stated as:

$$\begin{aligned} &\text{maximize} && \Phi_1 = \left\{\phi_s\right\}, \\ &\text{and minimize} && \Phi_2 = \left\{\phi_v, \phi_f\right\}, \\ &\text{such that} && \psi_c = 0, \end{aligned}$$

with ϕ_s defined in Equation (4.14). In other words, the thermal optimization problem investigated in this application consists in designing the embedded network to achieve optimal heat exchange between the microvascular fin and its surrounding.

Flow boundary conditions on opposite sides

The Pareto-optimal surface for this case is given in Figure 4.14. For this problem, the surface convected energy objective function ϕ_s is maximized, thus the Pareto-optimal front has a different configuration when compared with the fronts of previous problems.

Moreover, this front contains a feature that is particularly different, for the candidate solution that minimizes the void volume fraction objective function, labeled (a) in the front, is an inflection point for the ϕ_s objective function. Starting from this individual, solutions with increasing values of ϕ_f create a trail of individuals that have increasingly higher values of ϕ_s. The solution located at the extremum of this trail, even though not being optimal in any of the objective functions, has also been selected for visualization to explain this particular feature of the front.

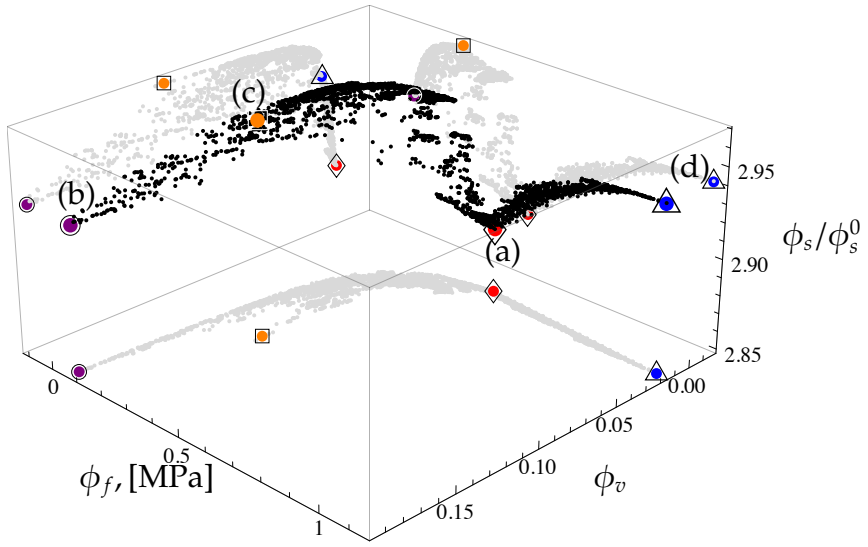


Figure 4.14 – NSGA-II results at generation $t = 10000$ for the fin specimen example with flow boundary conditions on opposite sides of the domain. Selected candidate solutions, labeled in the front and marked with ◇, ◯, □, and △ symbols, are illustrated in Figure 4.15.

Figure 4.15 presents the selected candidate solutions for this problem. The individuals that minimize ϕ_v and ϕ_f, shown in Figures 4.15a and 4.15b, have similar features as those shown in the previous problems. These figures are labeled (a) and (b) in the Pareto-optimal front of Figure 4.14, and are marked with ◇ and ◯ symbols, respectively. The solution presented in Figure 4.15c, labeled (c) in the front and marked with a □ symbol, maximizes the surface convected energy objective function ϕ_s. Therefore, this individual maximizes the difference $u^h - u_\infty$ over the entire domain Ω^h, and because u_∞ is constant, this means that the temperature is minimized on average over the domain. A similar solution would thus be obtained by considering the average temperature of the domain as an objective function, instead of using the defined objective function ϕ_s. The microvascular network for this solution roughly distributes the flow evenly through the surface of the specimen. First, the network halves the flow and channels it to the top and bottom edges of the specimen.

Then, part of the flow remains in these edges, while the rest is channeled back to the center of the specimen. Near the outflow, the flow is collected again near the top and bottom edges in two halves before exiting the domain. The last solution shown in Figure 4.15d is presented to explain the trail in the Pareto-optimal surface shown in Figure 4.14, a feature that was not present in the previous optimization examples. This individual is marked with a $\triangle$ symbol and is labeled (d) on the front. The individual shows that, starting from the individual with minimum void volume fraction, better configurations for the convected surface energy can be obtained with networks that collect the flow into a single set of interconnected microchannels. Increasing the length of these single-path networks increases the pressure drop, as determined by Equation (4.2), and thus reduces the flow efficiency. However, the flow is exposed to more surface area, which has the effect of reducing the temperature more efficiently and thus increasing the difference $u^h - u_\infty$ over Ω^h. This serpentine-like topology therefore seems to be a good candidate solution for minimizing the temperature of the domain, to the cost of decreasing the flow efficiency. Note however, that this solution is worse than that shown in Figure 4.15c with respect to ϕ_s.

Flow boundary conditions on the same side

For this last optimization problem, the inflow and outflow are located on the left edge of the domain. Again, after 10000 generations, the Pareto-optimal surface for this problem is well established, as displayed in Figure 4.16. Once more, the selected individuals marked on the front are displayed in Figure 4.17. Figures 4.17a and 4.17b follow the same structure as the corresponding solutions in previous optimizations. The solution illustrated in Figure 4.17c maximizes the value of the objective function ϕ_s. The flow is entirely collected close to the inflow and outflow locations. Close to the right edge, the flow is divided in many paths, thus reducing the pressure drop in this region. The last figure shows the individual with the worst value of the flow efficiency objective function. This individual has a similar structure than the one shown in Figure 4.17c close to inlet and outlet locations. However, because the flow is collected into a single path of channels, the pressure drop is increased dramatically. Still, this individual is optimal for the surface convected energy objective function ϕ_s.

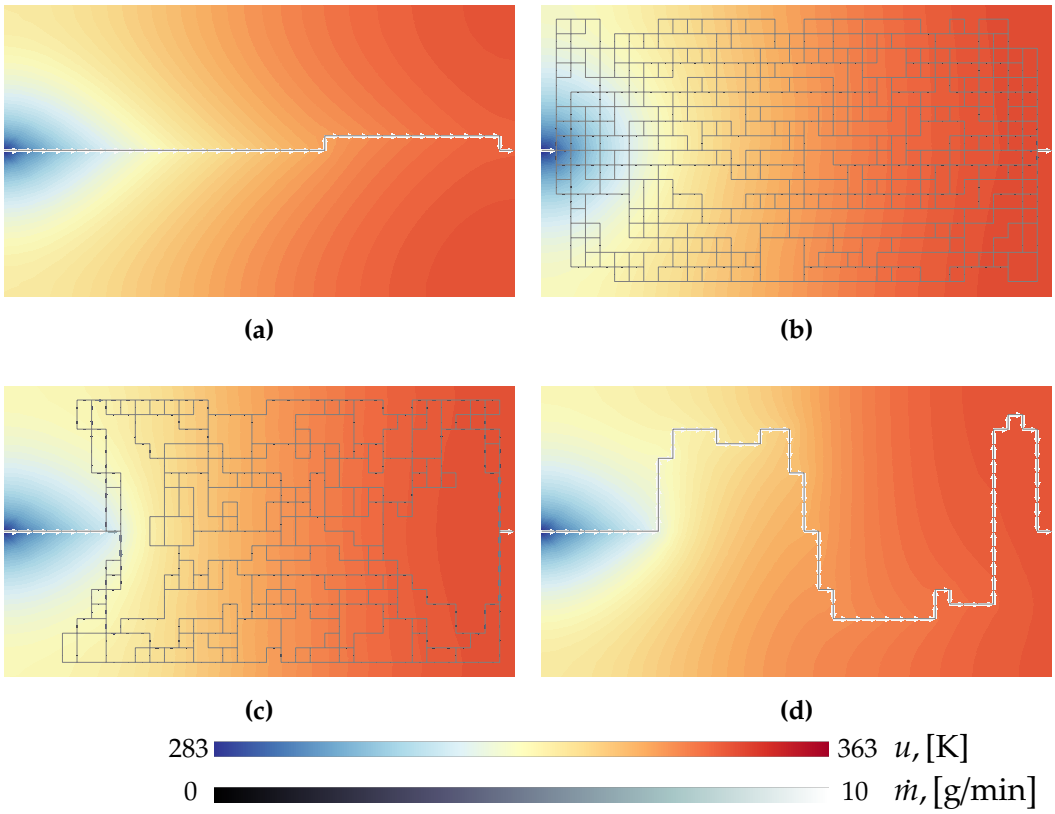


Figure 4.15 – Selected candidate solutions for the fin example considering constant inflow and flow boundary conditions on opposite sides of the domain. Figures (a), (b) correspond to the candidate solutions that minimize ϕ_v and ϕ_f, respectively. Figure (c) corresponds to the individual that maximizes ϕ_s. Finally, the individual illustrated in Figure (d) is displayed to explain the trail in the Pareto-optimal surface of Figure 4.14.

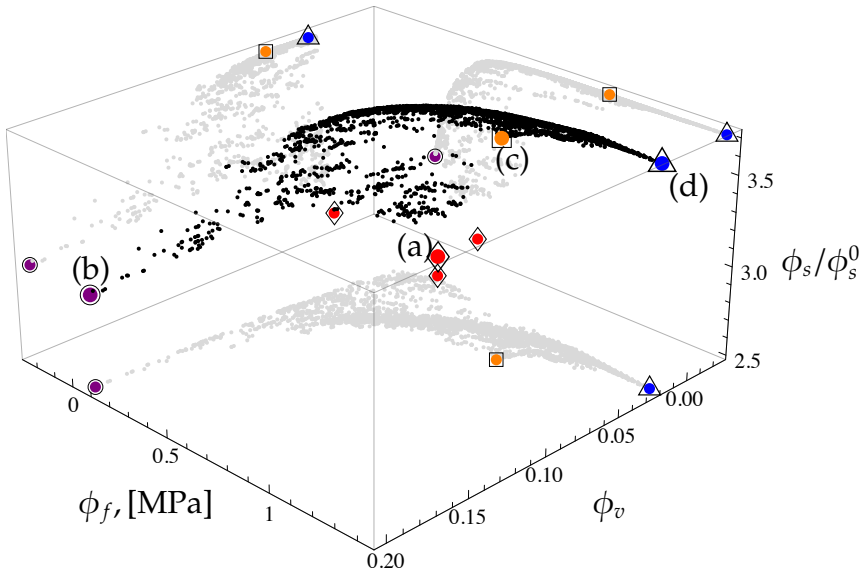


Figure 4.16 – Pareto-optimal surface at generation $t = 10000$ for the fin specimen example with the inflow and outflow located on the same edge. Individuals selected for visualization are displayed in Figure 4.17 and are labeled in the front and marked with symbols ◇, ◯, □, and △ symbols.

4.7 Conclusions

This work has presented a multi-objective constrained optimization framework for the design of microvascular networks for active cooling applications. The scheme is able to accurately define Pareto-optimal fronts for the objective functions considered, thus capturing the trade-offs between competing objectives for a final design decision. The algorithm can be used with an arbitrary number of objective functions and constraints. It was shown how coupling a finite element PDE solver with the multi-objective constrained genetic algorithm provides a powerful tool for multi-physics optimization. It is worth mentioning that even though extreme individuals on the Pareto-optimal fronts have been reported, the entire front is available to the analyst for a final decision.

The initial population usually contains only unfeasible individuals, but the population becomes feasible after only a few generations. There is a considerable difference between the evaluation times of feasible and unfeasible individuals. Strictly speaking, unfeasible individuals are not even evaluated, their objective functions are penalized with very high values. This difference in evaluation times requires a very efficient parallel implementation, that takes into account the load balancing on processors to achieve optimal speed-ups.

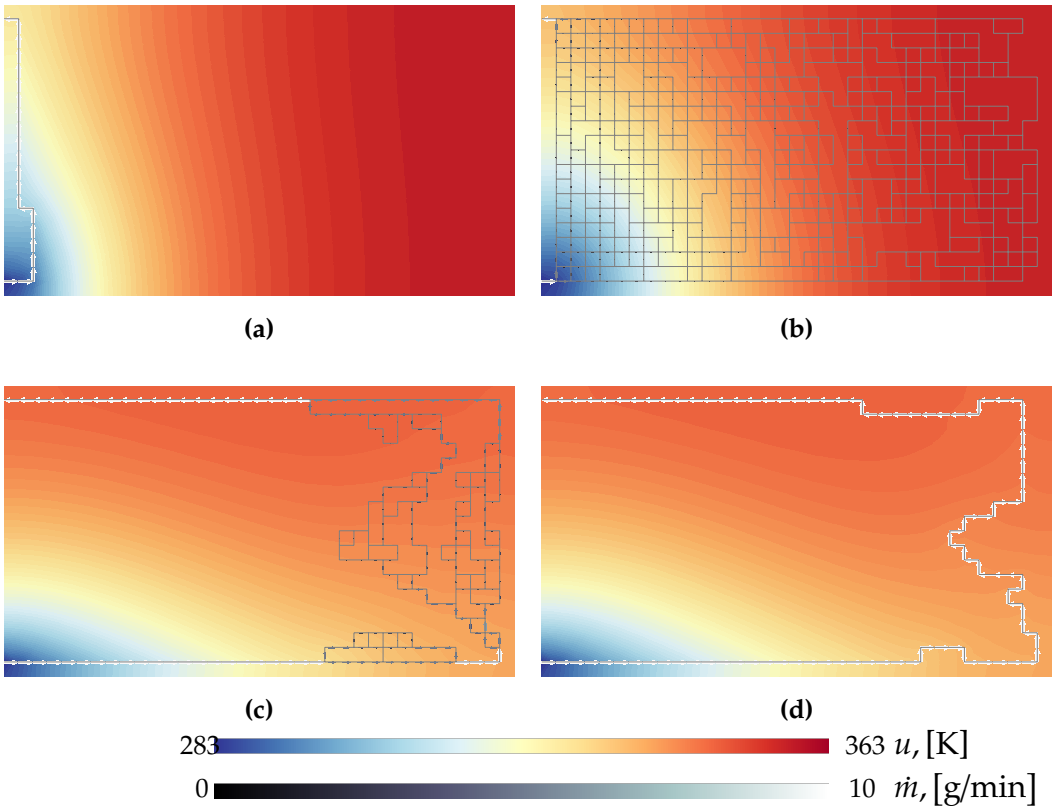


Figure 4.17 – Selected individuals taken from the Pareto-optimal front of Figure 4.16 for the microvascular fin example with the inflow and outflow located on the same edge of the domain. As before, Figures (a) and (b) minimize objective functions ϕ_v and ϕ_f, respectively. Figures (c) and (d) show candidate solutions with a high value of ϕ_s.

5 3D multi-physics optimization

5.1 Introduction

In this final chapter, we extend the 2D network design analysis presented in Chapter 4 to more complex 3D geometries. As before, the optimization of the network topology relies on a combination of the multi-objective constrained genetic algorithm summarized in Chapter 2 with a thermal finite element solver. The latter is used to capture the impact of the embedded network on the thermal response of the polymeric material. The optimization of three-dimensional polymeric components is driven by several motives. First, these polymeric materials are three-dimensional by nature, even when the ink scaffold is created layer by layer. Moreover, new advances in the manufacturing process of these materials allow for the creation of fairly complicated three-dimensional network topologies with the aid of a gel structure that supports the ink manifold, as explained in Chapter 1. A 3D mathematical model is thus required to keep up with the manufacturing advances. Second, there are several limitations imposed by a 2D model on the heat transfer response of these materials. In Chapter 4, the use of a temperature-dependent heat source helped circumvent the limitation of a 2D model to take into account the convection between the surface of the specimen and its surrounding environment. A 3D model does not have this limitation, as every surface can have a convective boundary condition. Furthermore, a 3D model can also capture the heat conduction through the thickness of the polymeric component. Due to the structured nature of the templates used hereafter to design the networks, the thermal solver does not rely on the GFEM. Nevertheless, the thermal solver relies on assumptions similar to those adopted in Chapter 4 to model the cooling of the embedded microchannels.

This chapter is organized as follows: Section 5.2 states the optimization problem and presents the mathematical model used for representing the microvascular networks. Section 5.3 presents a brief overview of GAs. The objective functions and constraints used in conjunction with the GA are defined in Section 5.4. Section 5.5 summarizes a validation study of the thermal field when these materials are used for active cooling applications. Finally, the application of the optimization framework for the design of 3D microvascular

networks follows in Section 5.6.

5.2 Microvascular network representation

Consider the mathematical model of a biomimetic material shown in Figure 5.1. An open domain $\Omega \subset \mathbb{R}^3$, with boundary $\Gamma := \overline{\Omega} - \Omega$, represents the polymeric component. The optimization process starts from what we call the microvascular *network template*, a reference structure that contains every possible microchannel considered in the design. Mathematically, the network template is represented by a graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where each edge $e \in \mathcal{E}$ represents a channel of the network and each vertex $v \in \mathcal{V}$ represents the possible location of a microchannel endpoint. Thus, a space coordinate $\mathbf{x}_v \subset \Lambda \subset \Omega$ is associated with each vertex v, where $\Lambda := \{\sum_{i=1} z_i \mathbf{e}_i \mid z_i \in \mathbb{Z}\}$ is a three-dimensional point lattice and $\mathbf{e}_i$ represents the $i-$th component of an orthogonal basis of $\mathbb{R}^3$. Restricting the location of the vertices to a point lattice reduces considerably the number of possible network topologies within Ω. Other properties assigned to vertices include pressure $p_v : v(\mathcal{V}) \to \mathbb{R}$ and mass flow rate $\dot{m}_v : v(\mathcal{S} \cup \mathcal{T}) \to \mathbb{R}$, $\mathcal{S} \cap \mathcal{T} = \emptyset$, where $\mathcal{S}$ and $\mathcal{T}$ represent inflow and outflow vertex sets, i.e., $\{\mathcal{S}, \mathcal{T}\} \subset \mathcal{V}(\mathcal{G})$. Because of mass conservation, the total mass inflow rate $\dot{m}_\mathcal{S} := \sum_{v \in \mathcal{S}} \dot{m}_v$ equals the total mass outflow rate $\dot{m}_\mathcal{T} := \sum_{v \in \mathcal{T}} \dot{m}_v$. Properties are also assigned to the edges of the graph, including the microchannel diameter $D_e : e(\mathcal{E}) \to \mathbb{R}$, length $L_e : e(\mathcal{E}) \to \mathbb{R}$, and mass flow rate $\dot{m}_e : e(\mathcal{E}) \to \mathbb{R}$.

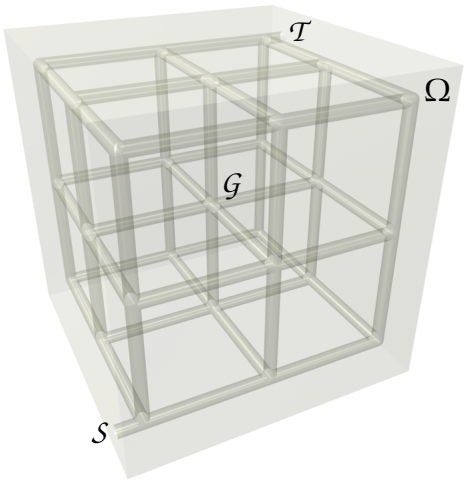


Figure 5.1 – Mathematical model of a biomimetic polymeric component. An embedded microvascular network template is represented by a graph $\mathcal{G}$, with inflow and outflow vertices $\mathcal{S}$ and $\mathcal{T}$, respectively.

As explained in the previous chapters, these microvascular materials are created by extruding a fugitive ink on a polymeric substrate. Let $\mathcal{D} = \{D_i\}_{i=1}^{k}$ be the set of discrete diameters used in the optimization process. This set includes the diameters prescribed by the manufacturing process, i.e., the diameter of the nozzles used to extrude the fugitive ink. It also includes $D_k = 0$, which represents the absence of a microchannel to allow networks to change in topology from that of the network template. During the optimization process, each microchannel $e \in \mathcal{E}$ of a candidate solution is assigned a diameter $D_e \in \mathcal{D}$ in order to optimize a set of objective functions $\Phi = \left\{\phi_i\right\}_{i=1}^{m}$, subject to a set of constraints $\Psi = \{\psi_i\}_{i=1}^{n}$. Each candidate solution created during the GA optimization process can be thought of as a mapping, a function $f : \mathcal{G} \rightarrow \mathcal{G}' \subseteq \mathcal{G}$, where $\mathcal{G}' := (\mathcal{V}', \mathcal{E}')$ may contain fewer channels, i.e., $\mathcal{E}' \subseteq \mathcal{E}$. Thus, the *degree* of a vertex v' in the candidate solution, denoted $\deg(v')$ and defined as the number of incident edges on the vertex, may be smaller than the degree of the corresponding vertex in the network template $\mathcal{G}$, i.e., $\deg(v' \in \mathcal{V}') \leq \deg(v \in \mathcal{V})$. It is interesting to note that there are $k^{|\mathcal{E}|}$ possible network configurations in this optimization process. Yet many of these topologies are considered unfeasible because their corresponding graphs are *disconnected*, which means that there is at least one pair of vertices $v_i', v_j' \in \mathcal{V}'(\mathcal{G}')$ for which a path cannot be created through the existing edges $\mathcal{E}'(\mathcal{G}')$. A disconnected graph therefore contains two or more *connected components*.

5.3 Multi-objective constrained genetic algorithms

Genetic algorithms are chosen as the underlying optimization technique because of their advantages with regard to other methods used in search and optimization. As a subset of a broader category of Evolutionary Algorithms (EAs), genetic algorithms are population-based methods where candidate solutions develop towards better configurations for the problem considered throughout an evolution process. GAs do not require any information about optimal topologies, a main advantage when comparing the method with other approaches used for optimization such as constructal theory [11, 12, 102]. Usually, the starting population of a GA is created randomly, yet any *a priori* knowledge about optimal features can be incorporated into this initial population, procedure known as a knowledge-based GA. After the creation of the initial population, GAs usually start the evolution over a specified number of generations $t_{\max}$, unless other criteria can be used to obtain a measure of convergence. More precisely, GAs can evolve indefinitely introducing new features to individuals in the population. GAs do not guarantee the convergence to optimal solutions at any value of $t_{\max}$, but they search the entire decision space, a main advantage

when comparing GAs to gradient-based search techniques like topology optimization [24, 106, 15]. Also, GAs do not require any gradient computation, and they can be used as long as a quantitative measure of the fitness can be assigned to the individuals in the population. Additionally, GAs can handle problems that are encoded with both discrete and continuous variables. As explained in Section 5.2, the optimization problem in this work assigns discrete values to the diameters of the microchannels of candidate solutions, making this a discrete optimization problem.

At each generation t of the evolution of the GA, the genetic operators *selection*, *crossover* and *mutation* are applied to generate the population at generation $t + 1$. The selection operator can be thought of as a mechanism that eliminates those individuals with low fitness so that they are excluded in successive generations, thus fulfilling Darwin's natural selection. Through the crossover operator, selected individuals exchange their features to produce new individuals. As a result, the created individuals contain features found in their parents. Finally, the mutation operator slightly modifies the characteristics of the newly created individuals and can bring new features not present in their parents nor in the entire population. In this work, tournament selection, uniform crossover and bitwise mutation are used as genetic operators [16, 17]. It is worth noting that bitwise mutation can only be applied if $k = 2$. If more diameters are considered, Gaussian mutation can be used as the mutation operator (see Chapter 2). The simple GA [16] employs only the genetic operators mentioned above to determine the population at generation $t+1$. However, when dealing with multiple objective functions, it is better to use a Pareto selection mechanism to determine the population at generation $t+1$. Through this procedure, individuals are compared in all objective functions simultaneously, and *dominated* candidates are likely to be discarded in subsequent generations. When comparing two contenders, the dominated individual is worse in at least one objective function while not being better in the rest. The concept of dominance is due to Vilfredo Pareto [45] from the field of economics.

Having the genetic operators for the creation of new candidate solutions and the Pareto selection mechanism to deal with multiple objectives, the remaining challenge involves the consideration of constraints. The explanation given in the previous paragraph deals with the objective function space. Similarly, we can apply the same ideas to a space dealing with constraints, thus better individuals can be thought of being "less unfeasible". However, this approach works only when dealing with unfeasible individuals. If at a particular generation t, the population is composed of both feasible and unfeasible candidates, the strategy adopted in this work to determine the better of two contenders is as follows: If one individual is feasible and the other is not, the feasible individual is selected. If

both individuals are unfeasible, select the less unfeasible one based on the values of the constraints. Finally, if both individuals are feasible or if an individual cannot be selected because they have the same constraint values, select the best individual in the objective function space. In this work, the Nondominated Sorting Genetic Algorithm II (NSGA-II) [46] with the addition of a constraint space is used as the multi-objective constrained genetic algorithm. For more information, refer to Chapter 2.

The evolution of a GA requires a large number of evaluations, thus obtaining meaningful results can be computationally demanding. However, the evaluation part of the GA is *embarrassingly parallel*, which means that candidate solutions can be evaluated in different processes independently. This property of the GAs make them suitable for distributed-memory machines. The algorithm used in this work adopts a master-slave approach, where the master process sends the individuals so that they can be evaluated independently by the slave processes, and then the master process collects the results afterwards. Close to ideal speed-ups are obtained using this approach. A detailed description of the parallelization scheme adopted in this work is given in Chapter 4.

5.4 Design objectives and constraints

This section describes the objective functions and constraints taken into account in the 3D microvascular network design problem.

5.4.1 Void volume fraction

The void left by the microvascular network has a direct impact in the mechanical properties of the resulting polymer. It is desired to keep the void volume fraction to a minimum in order to reduce the network's impact on the stiffness and strength of the material. The void volume fraction objective function is defined as a function $\phi_v : \mathcal{E}'(\mathcal{G}') \rightarrow \mathbb{R}$:

$$\phi_v := \pi \frac{\sum_{e' \in \mathcal{E}'} L_{e'} D_{e'}^2}{4 V_\Omega}, \tag{5.1}$$

where the volume of the network is normalized by the volume of the polymeric component V_Ω.

5.4.2 Flow efficiency

Flow efficient networks will have a configuration such that the energy needed to drive the fluid through the microchannels is kept to a minimum. The quantitative evaluation of the flow efficiency is carried out through the computation of the maximum pressure drop in the network. Assuming laminar flow, the pressure drop in microchannel $e' \in \mathcal{E}'$ can be expressed through the Hagen-Poiseuille's law:

$$\Delta p_{e'} = \frac{128 \nu L_{e'}}{\pi D_{e'}^4} \dot{m}_{e'}, \tag{5.2}$$

where ν is the kinematic viscosity of the fluid. Using Equation (5.2) to express the mass flow rate at microchannel endpoints gives a total of $2|\mathcal{E}'|$ equations, which are used to assemble the system of linear equations $\vec{K}\vec{p} = \vec{m}$. In this system, $\vec{K}$ is the network characteristic matrix, $\vec{p}$ is the network pressure vector and $\vec{m}$ is the network mass flow rate vector. This system is solved after applying flow boundary conditions, which include prescribed mass flow rate at inflow vertices $v' \in \mathcal{S}$ and prescribed pressure at outflow vertices $v' \in \mathcal{T}$. Then the flow efficiency objective function is defined as a function $\phi_f : \mathcal{V}'(\mathcal{G}') \to \mathbb{R}$ such that

$$\phi_f := \max_{v'_i, v'_j \in \mathcal{V}''} |p_{v'_i} - p_{v'_j}|. \tag{5.3}$$

More precisely, the computation of the maximum pressure drop is performed in each connected component of the graph $\mathcal{G}'$, each one with its own flow boundary conditions. That is, it is possible that during the mapping process that creates a new candidate solution, a disconnected network with two or more connected components is created. As long as those connected components have valid boundary conditions for the flow problem, the networks are considered as feasible and the computation of the maximum pressure drop is carried out component by component.

5.4.3 Thermal field

By obtaining the temperature field in the polymer, objective functions can be defined to obtain highly efficient networks for active cooling applications. Consider the boundary of the domain Γ with outward normal $\mathbf{n}$, partitioned in regions Γ_u, Γ_q, and Γ_h, which correspond to regions with applied Dirichlet, Neumann, and Robin boundary conditions, respectively. These regions are mutually exclusive, thus $\Gamma_u \cap \Gamma_q \cap \Gamma_h = \emptyset$. Also, the closure of their union defines the entire boundary of the polymeric compound, i.e., $\Gamma = \overline{\Gamma_u \cup \Gamma_q \cup \Gamma_h}$.

It can be shown through an energy balance [90], that the heat rate per unit length q_e in the wall of microchannel e is given by

$$q_e = \dot{m}_e c_p \frac{du}{d\xi_e}, \tag{5.4}$$

where u denotes temperature, c_p the specific heat of the fluid, and ξ_e the local axial coordinate of the channel. By adopting Equation (5.4), several simplifying assumptions have been made, including a fully developed temperature profile and a constant heat rate along the wall of the channel. Network microchannels can be collapsed to 1D line segments due to their high aspect ratios and the small diameters used. Thus, Equation (5.4) allows us to consider the cooling effect of the microchannels as heat sinks over their collapsed lines. The steady-state thermal boundary value problem is given as follows: Given fluid density ρ and fluid specific heat c_p constants, ambient fluid temperature u_∞, the thermal conductivity tensor $\boldsymbol{\kappa} : \overline{\Omega} \to \mathbb{R}^3 \times \mathbb{R}^3$, the heat source $f : \Omega \to \mathbb{R}$, heat transfer coefficient $h : \Gamma_h \to \mathbb{R}$, and prescribed heat flux $\bar{q} : \Gamma_q \to \mathbb{R}$, find the temperature field $u : \overline{\Omega} \to \mathbb{R}$ such that

$$\nabla \cdot (\boldsymbol{\kappa} \nabla u) + f = \delta (\mathbf{x} - \mathbf{x}_e)\, c_p \dot{\mathbf{m}}_e \cdot \nabla u \qquad \text{on } \Omega, \tag{5.5}$$

with $\dot{\mathbf{m}}_e = \dot{m}_e \mathbf{e}_{\xi_e}$, and boundary conditions

$$u = \bar{u} \qquad \text{on } \Gamma_u, \tag{5.6}$$

$$\boldsymbol{\kappa} \nabla u \cdot \mathbf{n} = \bar{q} \qquad \text{on } \Gamma_q, \tag{5.7}$$

$$\boldsymbol{\kappa} \nabla u \cdot \mathbf{n} = h\,(u_\infty - u) \qquad \text{on } \Gamma_h. \tag{5.8}$$

In Equation (5.5), the Dirac delta function allows us to represent the 1D microchannels within the 3D computational domain.

For the weak formulation, let $\mathcal{U} = \{u \mid u|_{\Gamma^u} = \bar{u}\} \subset H^1(\overline{\Omega})$ be the set of trial solutions for the temperature field and $\mathcal{V} = \{v \mid v|_{\Gamma^u} = 0\} \subset H^1_0(\overline{\Omega})$ be the variation space. The weak form of the problem is: Given ρ, c_p, u_∞, f, h, $\boldsymbol{\kappa}$, $\bar{q}$, $\bar{u}$ as before, find $u \in \mathcal{U}$ such that

$$a\,(w, u) + a\,(w, u)_{\Gamma_h} = (w, f) + (w, \bar{q})_{\Gamma_q} + (w, u_\infty)_{\Gamma_h} \qquad \forall w \in \mathcal{V}, \tag{5.9}$$

with bilinear and linear forms

$$a(w,u) = \int_{\Omega} \left(\nabla w \cdot (\boldsymbol{\kappa} \nabla u) + w\, \delta(\mathbf{x} - \mathbf{x}_e)\, c_p \dot{\mathbf{m}}_e \cdot \nabla u \right) d\Omega, \tag{5.10}$$

$$a(w,u)_{\Gamma_h} = \int_{\Gamma_h} whu \, d\Gamma, \tag{5.11}$$

$$(w,f) = \int_{\Omega} wf \, d\Omega, \tag{5.12}$$

$$(w,\bar{q})_{\Gamma_q} = \int_{\Gamma_q} w\bar{q} \, d\Gamma, \tag{5.13}$$

$$(w,u_\infty)_{\Gamma_h} = \int_{\Gamma_h} whu_\infty \, d\Gamma. \tag{5.14}$$

An approximate temperature distribution in the material $u^h(\mathbf{x})$ is obtained by solving Equation (5.9) with a finite element discretization of the domain Ω^h. Various thermal efficiency objective functions can then be defined, such as the average temperature over the polymeric component, the thermal energy stored, or even the temperature gradients if the stresses produced by the thermal field are of interest. In the examples presented in this chapter, we adopt the maximum temperature objective function $\phi_T : \Omega^h \to \mathbb{R}$:

$$\phi_T := \max u^h(\mathbf{x}), \qquad \forall \mathbf{x} \in \Omega^h. \tag{5.15}$$

5.4.4 Connectivity constraint

The objective functions introduced above are computed only for connected graphs. When the candidate solutions are allowed to change in structure throughout the optimization process, many of the subgraphs $\mathcal{G}'$ of the graph $\mathcal{G}$ that represents the network template become unfeasible. When a solution to the flow problem cannot be computed, the candidate solution is marked as unfeasible. The connectivity constraint is defined as a function $\psi_c : \mathcal{E}'(\mathcal{G}') \to \mathbb{R}$ such that

$$\psi_c := \begin{cases} 1/\left|\mathcal{E}'(\mathcal{G}')\right| & \text{if } \mathcal{G}' \text{ is disconnected,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.16}$$

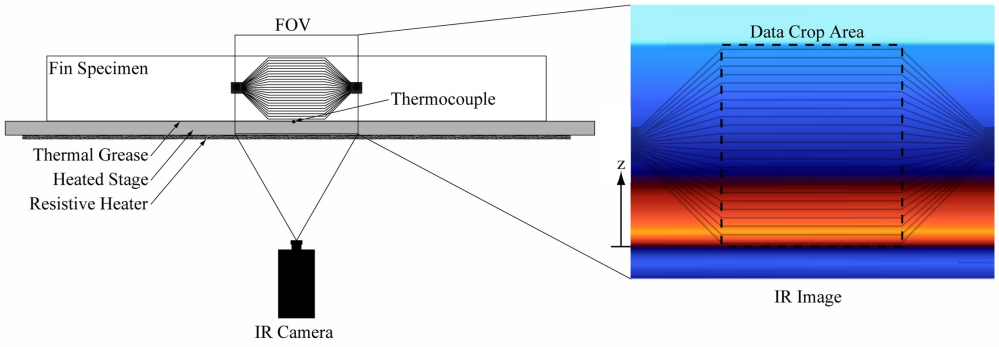
Because ψ_c is inversely proportional to the number of edges in the candidate solution, the optimization process pushes unfeasible individuals towards a known valid topology, i.e., that of the microvascular network template.

5.5 Validation study

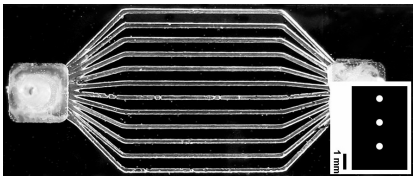
The numeric solution of Equation (5.9) is now compared with experimental measurements obtained by B. Kozola, R. Saksena, and V. Natrajan at the University of Illinois. Three sets of experimental results are used in this study: *a)* the surface temperature of the polymeric component obtained by infrared (IR) imaging of the specimen [97]; *b)* the temperature of the fluid through the two-dye LIF (Laser Induced Fluorescence) technique [110]; and *c)* the mass flow rate in the microchannels via the microscopic Particle Image Velocimetry ($\mu-$PIV) technique [108]. It is worth mentioning that the experimental results used in this validation study for the infrared imaging of the polymer surface and the two-dye LIF were obtained almost simultaneously, whereas mass flow rates were measured at a different time.

The measurement of the temperature field in the polymer is accomplished by obtaining images of the specimen with the aid of an IR camera. Figure 5.2a shows the experimental setup with the base of the specimen placed on a copper plate, which in turn lies on a resistive heater that is kept at constant temperature using a feedback controller. Thermally conductive grease is laid along the interface between the copper plate and the fin. The IR camera is utilized to capture the temperature distribution in the surface of the fin that contains the microvascular network. A detailed description of this technique can be found in [97]. The specimen used in the experiments is illustrated in Figure 5.2b. It consists of an epoxy polymeric fin with an embedded microvascular network containing twelve microchannels. Attachment nozzles are placed at the diverging and converging points of the fanning manifolds to provide the network with inlet and outlet locations. The microchannels have roughly a diameter $D_e = 410\,\mu$m. The height of the specimen is 19.5 mm, and its thickness is 3.5 mm. The portion of the fin where the microchannels are parallel to the heated stage measures 18.4 mm.

The mass flow rate in the individual microchannels is measured employing the $\mu-$PIV technique [108]. This method is able to determine the velocity field in a plane spanning the full diameter of a microchannel. The experimental setup for this technique, which mainly consists of a laser and a camera, is shown in Figure 5.3a. In order to obtain the velocity field in a channel, a solution of deionized water mixed with polystyrene microspheres containing a fluorescent red dye is driven through the network. By employing an optical filter, the camera captures the fluorescent light emitted from the particles. By cross-correlating two consecutive images taken by the camera at a specified time interval, the displacement (and therefore the velocity) field can be fully determined, and thereby the mass flow rate through the microchannel.

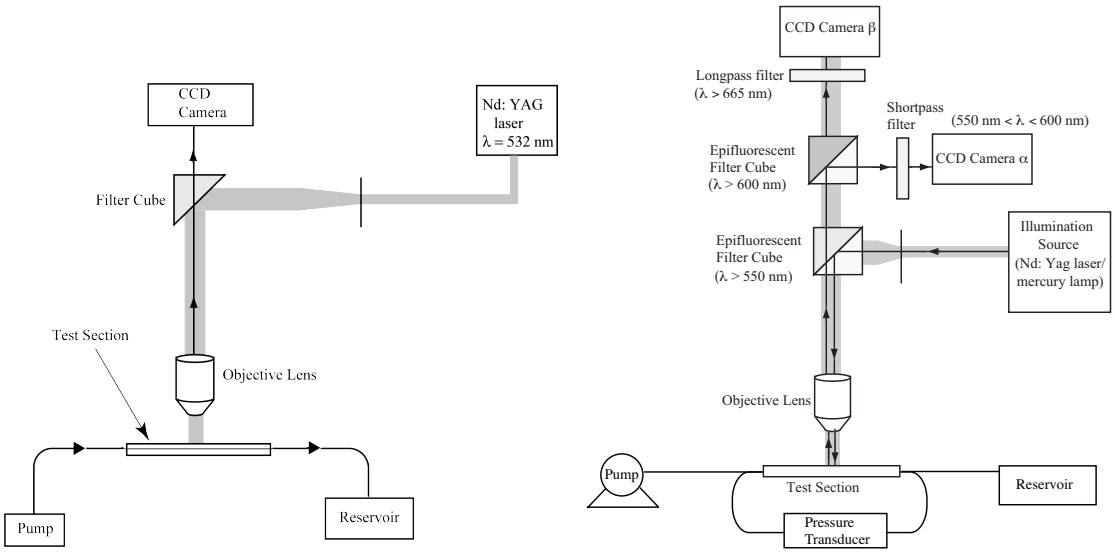


(a)



(b)

Figure 5.2 – (a) Experimental setup for the determination of the temperature field in the polymeric component. The figure shows the fin specimen lying on a copper plate that sits on a resistive heater. An infrared image of a sample in the absence of flow is illustrated on the right. (b) Fin specimen used in the validation study consisting of 12 microchannels that fan from inlet and outlet network locations.



(a) **(b)**

Figure 5.3 – (a) μ–PIV experimental setup; (b) LIF experimental setup.

Finally, the temperature of the fluid is determined by the two-dye LIF technique [110], with the setup illustrated in Figure 5.3b. The LIF employs a laser and two cameras with appropriate optical filters that are used to capture the emissions from their corresponding dye. One of the dyes is temperature sensitive, whereas the other is temperature insensitive. However, both dyes are sensitive to the incident energy of the laser, hence to remove the effect of variation in incident energy, the ratio of the images taken by the two cameras is used. The setup is calibrated against known temperatures in a water bath that is used to maintain constant temperature, which is additionally measured by a thermocouple.

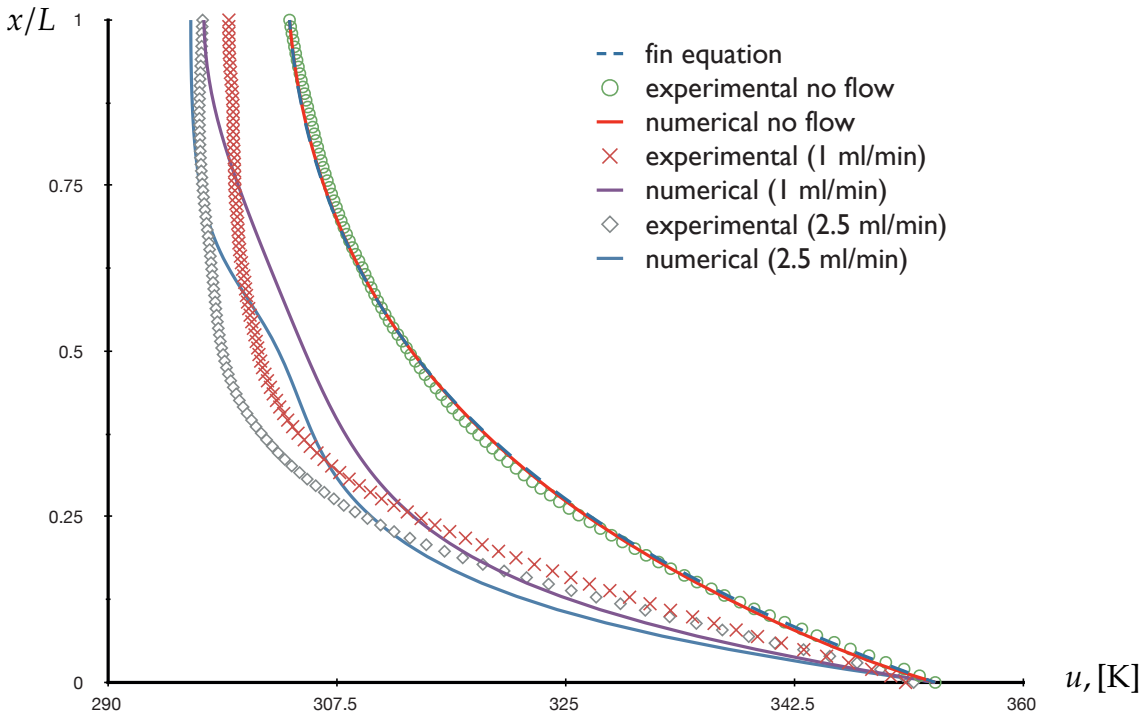
Figure 5.4 summarizes the results of the comparison between numerical predictions and experimental measurements of the temperature field for the specimen illustrated in Figure 5.2b. The abscissas show the surface temperature along the centerline of the specimen, and the ordinates the relative coordinate perpendicular to the heated stage. The symbols correspond to three experimental measurements. For the first experimental measurement, no flow was driven through the microvascular network. For the other two experimental results, a solution of deionized water with the two dyes used in the LIF technique was pumped through the twelve channels with total mass flow rates $\dot{m}_{s1}$ = 1 g/min and $\dot{m}_{s2}$ = 2.5 g/min, and the temperature measurements were obtained after achieving steady-state conditions. As expected, the temperature in the specimen drops with increasing values of the mass flow rate.

The *fin equation*, a simplified 1D equation that describes the energy balance that exists between the heat conducted through the fin and the heat convected to the surrounding fluid, can be used to obtain the film coefficient from experimental results. The equation has the form

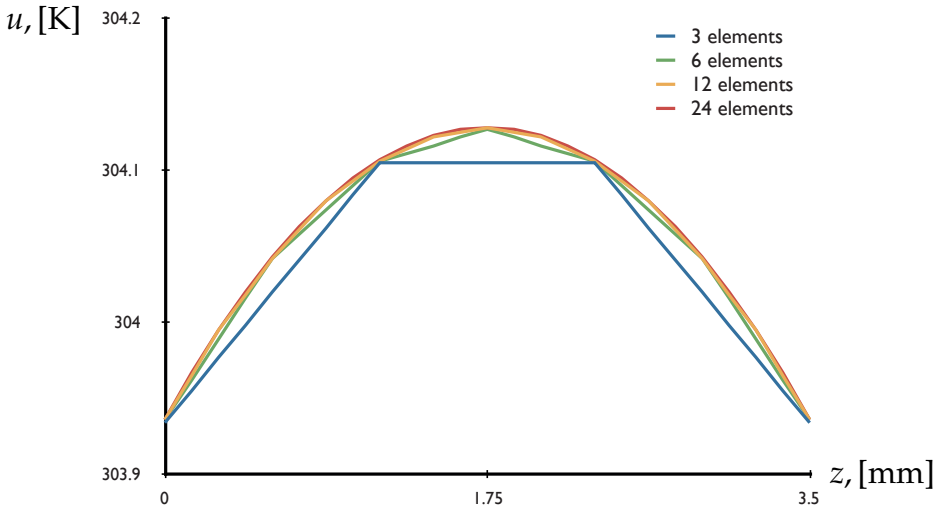
$$\frac{d^2u}{dx^2} = \frac{h_f P_f}{\kappa A_f}\left(u - u_\infty\right), \tag{5.17}$$

where P_f and A_f are the perimeter and the area, respectively, of the cross section in contact with the copper plate, x is the coordinate normal to this cross section, κ is the conductivity of the polymeric material and h_f is the heat transfer coefficient (or film coefficient) that characterizes the heat convection between the fin and the surrounding air with temperature u_∞. Boundary conditions include fixed temperature at the base of the fin (in contact with the copper plate) $u|_{x=0} = \bar{u}$, and a convective boundary condition at its free end (i.e., $x = L = 19.5\,\text{mm}$) $-\kappa\left(du/dx\right)|_{x=L} = h_f\left(u\left(L\right) - u_\infty\right)$. With these boundary conditions, the solution to the fin equation is given by

$$\frac{u\left(x\right) - u_\infty}{\bar{u} - u_\infty} = \frac{\cosh\beta\left(L - x\right) + \left(\frac{h_f}{\beta\kappa}\right)\sinh\beta\left(L - x\right)}{\cosh\beta L + \left(\frac{h_f}{\beta\kappa}\right)\sinh\beta L}, \qquad \beta = \sqrt{\frac{h_f P_f}{\kappa A_f}}. \tag{5.18}$$



(a)



(b)

Figure 5.4 – (a) Temperature profiles obtained at the centerline of the specimen shown in Figure 5.2b. Curves show the experimental temperature measurements obtained by infrared imaging (symbols), the numerical solutions (solid curves) and the results of the fin equation (dashed line). The heat transfer coefficients used in the fin equation and in the numerical computations were calibrated to match the experimental temperature distribution in the absence of flow. (b) Temperature profile through the thickness obtained numerically at the top of the specimen for meshes with different thickness discretizations.

With $\bar{u} = 353.3\,\text{K}$ and an environmental temperature $u_\infty = 296.4\,\text{K}$, it was found that a heat transfer coefficient $h_f = 8.131\,\text{W/m}^2\text{K}$ matches closely the experimental results, as shown by the corresponding curve in Figure 5.4a.

A finite element model of the central portion of the domain that contains the straight microchannels, i.e., $18.4 \times 19.5 \times 3.5\ \text{mm}^3$, is created to obtain the numerical temperature predictions. In the absence of flow, the boundary conditions for the computational model include a prescribed temperature $\bar{u} = 353.3\,\text{K}$ at the base of the model, and convective boundary conditions along the top and along the two major surfaces. Insulated boundary conditions, i.e., $\bar{q} = 0$, are applied to the remaining surfaces of the computational model. For the convective boundary conditions, the environmental temperature is the same as the one used for the fin equation, and a value $h = 8.736\,\text{W/m}^2\text{K}$ for the heat transfer coefficient was obtained by matching the measured experimental temperature at the top of the fin. Thus, in the absence of flow, the numerical results match closely those of the fin equation and the experimental results, as illustrated in Figure 5.4a.

A mesh convergence study is carried out by creating four structured meshes of eight-node brick elements over the 3D grids $\left\{40 \times 48 \times 2^i \cdot 3\right\}_{i=0}^{3}$. In other words, the four meshes have 40 elements along the length 18.4 mm, 48 elements along the length 19.5 mm and either 3, 6, 12, or 24 elements through the thickness of 3.5 mm. For the four meshes, measurements of the temperature across the thickness over the top of the specimen are reported in Figure 5.4b. The figure shows that considering six elements is enough to capture the thermal effects through the thickness of the specimen. It also illustrates that the temperature at the extreme sides of the specimen, which are the locations where the temperature measurements can be obtained experimentally by infrared imaging, does not vary much, even when only considering three elements across the thickness.

For the numerical predictions considering the effect of the flow through the microvascular network, the mass flow rate in each microchannel is predicted by taking into account the actual geometry of the specimen. In other words, each microchannel length is measured and its corresponding mass flow rate is obtained by Equation (5.2). Also, the temperature of the fluid at inlet locations is prescribed in the computational model with the values reported in Table 5.1, which were obtained experimentally by the two-dye LIF technique described above. The temperature values of the fluid at other inlet locations were linearly extrapolated from those given in the table. The numerical results that take into account the effect of the flow are also given in Figure 5.4. It can be seen that the numerical curves follow the same trend as the experimental ones, which means that the temperature reduces as the mass flow rate increases. Also, both numerical and experimental results share the same order of magnitude, but there are some discrepancies between

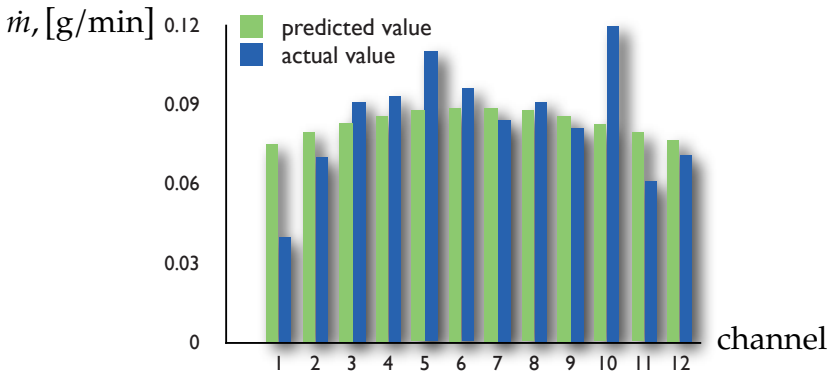
	mass flow rate $\dot{m}$,	[g/min]
channel	1	2.5
2	312.4	311.9
4	309.5	305.8
7	305.2	305.1
8	303.2	297.0

Table 5.1 – Fluid temperature values obtained by the two-dye LIF technique, given in Kelvin, measured at the start of the straight portion of four microchannels for the specimen illustrated in Figure 5.2b. Channel 2 is the closest channel to the heated stage.

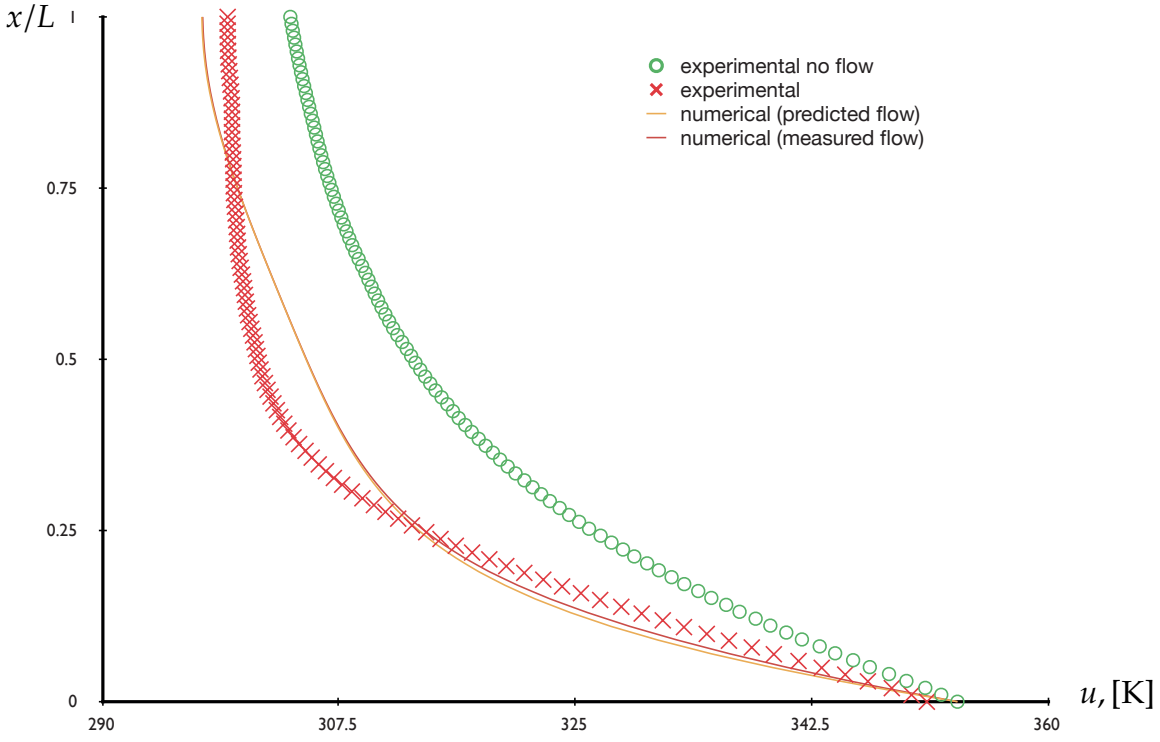
them. The first source of discrepancy is associated with the experimental error related to the temperature measurements at the inlet locations and with the interpolation/extrapolation of the temperature values given in Table 5.1 for the other eight microchannels. Another source of discrepancy is due to the departure of the predicted flow distribution across the twelve microchannels used for the numerical simulations from the actual flow field in the experiments. Indeed, as shown in Figure 5.5a, a substantial variability is observed experimentally (using the $\mu-$PIV technique), with some channels (especially the center ones) carrying more of the flow than others. Taking that variability into account in the numerical predictions leads to the results shown in Figure 5.5b. As apparent there, the differences observed in the temperature profile due to the variability in mass flow rate are negligible for the case $\dot{m}_{s1} = 1$ g/min. Therefore, the remaining discrepancy is probably associated with errors in the inflow temperature, emphasizing the need for the simultaneous measurement of the fin surface temperature and the inflow temperature of the cooling fluid in every microchannel.

5.6 Network optimization results

This section presents the results obtained by applying the optimization framework for the design of 3D microvascular polymeric components to two different applications. The first application consists of the design of a periodic microvascular cell that is part of a larger polymeric component, e.g., a thin wall structure used for high temperature applications. The second application presents another opportunity to compare with experimental temperature measurements and is motivated by the 2D optimization considering localized heating, discussed thoroughly in Chapter 4.



(a)



(b)

Figure 5.5 – (a) Predicted and actual mass flow rate values for each of the twelve channels of the specimen shown in Figure 5.2b for a total mass flow rate $\dot{m}_{s1}$ = 1 g/min. Actual values were obtained by using the μ−PIV technique. (b) Temperature profiles at the centerline of the fin specimen showing some of the results given in Figure 5.4a, and another numerical curve for which the mass flow rates were corrected with those given in (a). Results considering the flow in the network correspond to a total mass inflow rate $\dot{m}_{s1}$ = 1 g/min.

5.6.1 Periodic microvascular cell

Consider a cubic domain $\Omega = a^3$ with $a = 10$ mm, as displayed in Figure 5.6. The figure shows a part of the microvascular network template $\mathcal{G}$, which contains 1946 microchannels. With a diameter set $\mathcal{D}[\mu\text{m}] = \{0, 100\}$ used in the optimization process, there are 2^{1946} network configurations. As mentioned previously, even though many of these topologies are not feasible because they result in disconnected networks for the flow problem, the number of feasible structures is extremely large. The network template consists of channels aligned over an n^3 grid, with $n = 8$, with a total of $3n\,(n+1)^2$ microchannels, and two more microchannels to link inflow and outflow, all of them with length $L_e = 1$ mm. Thus, the network template roughly has a 1 mm cover from each of the six surfaces of the domain Ω. A single inflow and a single outflow are located at the bottom surface and are displayed using arrows in the figure. Flow boundary conditions comprise a prescribed mass inflow rate $\dot{m}_S = 1$ g/min at the inlet and zero pressure at the outlet. The fluid used as coolant is water, with kinematic viscosity $\nu = 1.05 \times 10^{-6}$ m^2/s and specific heat $c_p = 4185.5$ J/kg K. Results are reported only for constant inflow, so that the mass inflow rate does not change in all individual evaluations (see Chapter 4 for further details). For the heat boundary value problem, a thermal conductivity value of $\kappa = 0.3$ W/mK is used for the polymer. A uniform heat flux $\bar{q} = 3$ kW/m^2 is applied to the top surface and the bottom surface has a prescribed temperature $\bar{u} = 273.15$ K, which is also the temperature of the incoming fluid. Thus, in the absence of flow the temperature field has a linear variation in the z coordinate, with a maximum temperature $u_{\max} = 373$ K over the top surface. The remaining surfaces have a periodic boundary condition, thus defining a unit microvascular cell in the x and y directions, as displayed by the coordinate axes in the figure. The cubic domain is discretized with a structured finite element mesh, consisting of $20 \times 20 \times 20 = 8000$ eight-noded brick elements and a total of 9261 nodes. Note that a matching finite element mesh is used, thus the microchannels of the network template lie along the edges of finite elements in the mesh. By analyzing the grids used for the discretizing the domain and for outlining the microvascular network template, it can be inferred that each microchannel is surrounded by eight brick elements.

The problem is then optimized with a population size $|\mathcal{P}| = 1000$. The Pareto-optimal front for this problem is displayed in Figure 5.7 at generation $t = 10000$. Selected candidate solutions, labeled in the figure and marked with ◇, ◯, and □ symbols, are represented in Figure 5.8 for visualization. As with the results in Chapter 4, the front is projected into three orthogonal planes to ease its visualization. The projections show a well-defined surface, with some individuals isolated from the main population, e.g., those marked with ◇, ◯ symbols. In Figure 5.8, the upper row shows the temperature field, whereas the lower

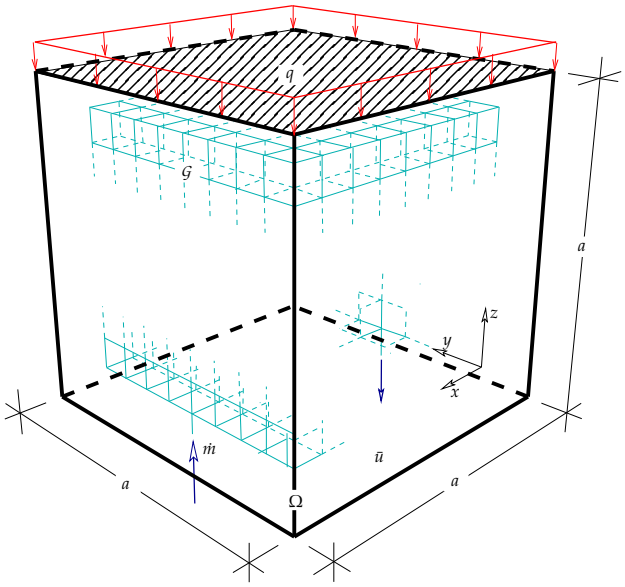


Figure 5.6 – Schematic of the periodic microvascular cell problem with cubic domain $\Omega = a^3$ and part of the microvascular network template $\mathcal{G}$. A single inflow and a single outflow are displayed in the figure with arrows. A uniform heat flux $\bar{q}$ is applied at the top surface, and a temperature $\bar{u}$ is prescribed at the bottom surface. The remaining surfaces have periodic boundary conditions, thus defining a unit cell periodic in the x and y directions.

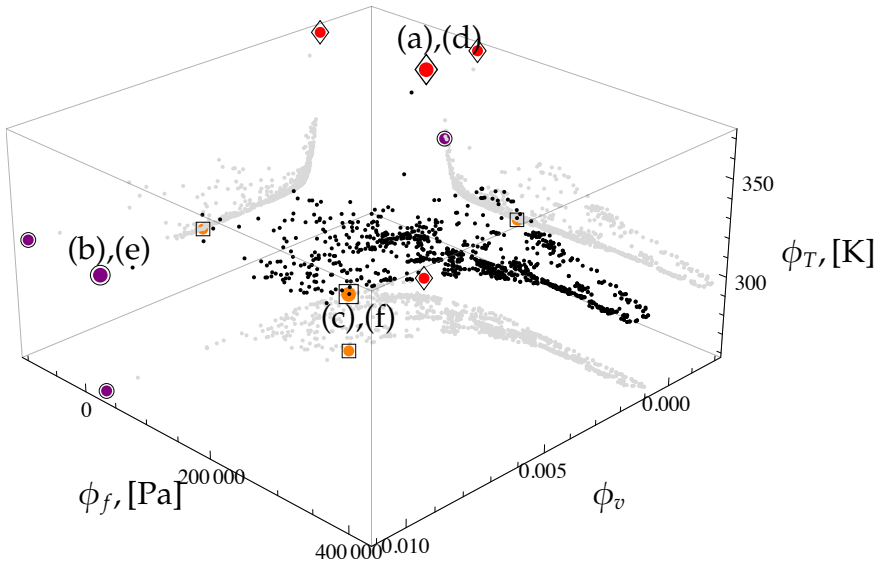


Figure 5.7 – Pareto-optimal surface at generation $t = 10000$ for the periodic cell example. Individuals selected for visualization are displayed in Figure 5.8 and are labeled in the front and marked with ◇, ◯, □ symbols.

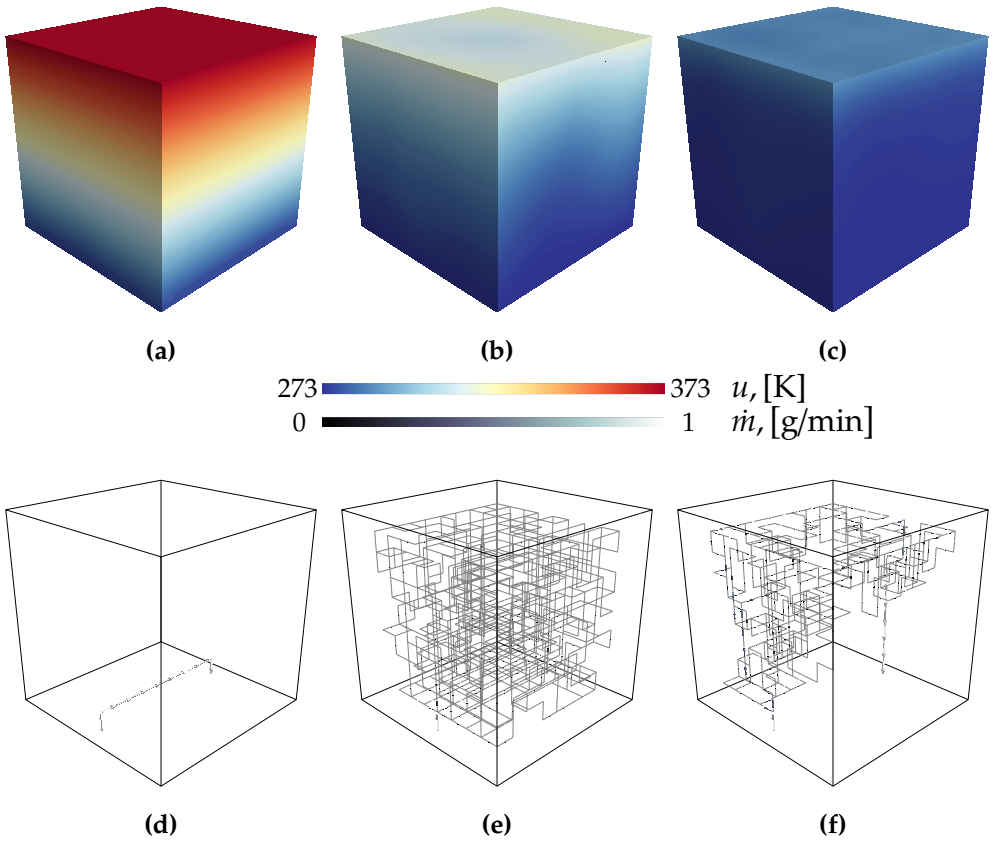


Figure 5.8 – Selected candidate solutions at generation $t = 10000$. The upper row shows the temperature field, whereas the figures in the lower row show the corresponding flow field with their corresponding network. Figures (a) and (d) correspond to the individual that minimizes ϕ_v, figures (b) and (e) to the individual that minimizes ϕ_f and figures (c) and (f) to the one that minimizes ϕ_T. These candidate solutions are shown in the Pareto-optimal front of Figure 5.7 with ◇, ◯, and □ symbols, respectively. Arrows are size-coded according to flow magnitude.

row shows the corresponding network and flow field. In the lower figures, the arrows that represent the flow are color-coded and scaled according to the flow magnitude.

The candidate solution that minimizes the value of ϕ_v, marked with a $\diamond$ symbol in Figure 5.7, is presented in Figures 5.8a and 5.8d. This is the most optimal solution for this objective function, thus the inlet and the outlet are connected using the least amount of void volume fraction, as revealed by Figure 5.8d. On the other hand, this solution is the worst for minimizing the maximum temperature objective function ϕ_T, reducing it by only 1 K from the configuration without flow. As explained previously, this individual is isolated in the front from the main population. The individual that maximizes the flow efficiency is shown in Figures 5.8b and 5.8e and is marked with a $\bigcirc$ symbol in the Pareto-optimal front of Figure 5.7. The network is very dense, so that its topology resembles that of the microvascular network template. After dividing the flow into four paths close to the inlet, the magnitude of the flow is reduced and rapidly becomes negligible close to the upper surface. The maximum temperature for this solution is reduced by approximately 54 K. Finally, the candidate solution that minimizes ϕ_T, corresponding to Figures 5.8c and 5.8f, is marked with a $\square$ symbol in the Pareto-optimal front of Figure 5.7. This individual reduces the maximum temperature by approximately 84 K. Figure 5.8f shows how the fluid is pushed from the inlet to the upper surface where the uniform heat flux is applied. Along the upper surface, the flow seems to be uniformly distributed. Then the flow is collected in the other side of the domain and taken to the outlet. Note the absence of microchannels in the interior of the domain, demonstrating that no network is required in this region to minimize the temperature. Referring back to Figure 5.7, it can be seen that this individual has a low value of the flow efficiency objective function ϕ_f. Dividing the flow in the upper surface has the effect of reducing the amount of energy needed to drive the flow, with the penalty of increasing the void volume fraction. Thus, this individual has also a hight value of ϕ_v.

It is worth mentioning that the candidate solutions that minimize ϕ_v and ϕ_f have a similar value of the flow efficiency objective function. Thus, for the dense network the division of the flow in several paths compensates the increase in length, thus maintaining a low value for the pressure drop between inlet and outlet locations.

5.6.2 Microvascular fin with localized heating

Consider the domain confined by the rectangular cuboid $\Omega = (36 \times 20 \times 4)\,\text{mm}^3$ shown in Figure 5.9. For this problem, the microvascular network template is aligned along a 2D $m \times n$ grid over the specimen's plane of symmetry that is parallel to the main

surfaces. Thus, the network has $m + n + 2mn$ channels, plus a couple used to connect with the flow boundary locations. With $m = 34$ and $n = 18$, there are a total of 1278

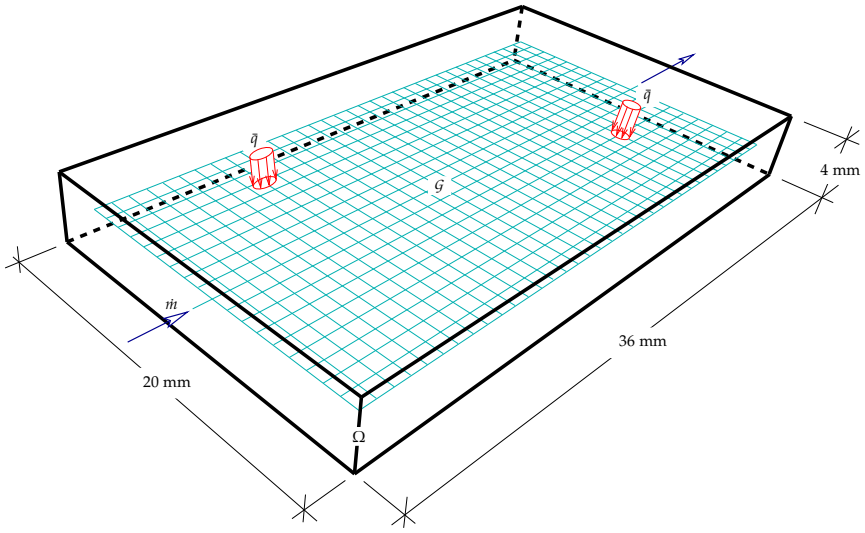


Figure 5.9 – Schematic of the mathematical model for the localized heating example, showing the domain Ω and the microvascular network template $\mathcal{G}$ used in the optimization process. Arrows show the location of the inflow and outflow for the flow problem. For the heat boundary value problem, a localized heat flux q is applied in two locations over the top surface.

microchannels. As in the previous problem, all microchannels have a length $L_e = 1$ mm, and the polymeric component has a cover of roughly 1 mm in each side over the plane of the network, and a cover of 2 mm in the direction perpendicular to this plane. A diameter set $\mathcal{D}[\mu\text{m}] = \{0, 100\}$ is used throughout the optimization process, resulting in 2^{1278} network configurations. For the flow boundary value problem, a single prescribed mass inflow rate of water $\dot{m}_S = 1\,\text{g/min}$ and zero pressure are considered as boundary conditions, and their locations are displayed in the figure with arrows. The coolant properties are the same as those given in the previous example. Regarding the solution of the thermal boundary value problem, the rectangular cuboid is discretized using a structured mesh of eight-node brick elements, with a total of $72 \times 40 \times 8 = 23040$ finite elements and 26937 nodes. Thus, this example contains a much larger number of elements compared to the periodic microvascular cell problem studied in the preceding example. A heat flux $\bar{q}$ is applied locally as displayed in the figure to the free surface of eight finite elements, i.e., to four elements per localized heat source. Convective boundary conditions, with ambient temperature $u_\infty = 293.15$ K and heat transfer coefficient h, are prescribed along every surface of the domain that does not contain the applied heat

fluxes. The numerical values of $\bar{q}$ and h are determined from experimental temperature measurements using the IR camera setup illustrated in Figure 5.2, from a manufactured polymeric specimen with the geometry outlined in Figure 5.9. The surface that contains the applied heat flux is considered insulated in the portions that do not contain the heat flux. The value of the prescribed inlet fluid temperature is $\bar{u} = 283.15\,\mathrm{K}$.

In the experimental setup, the localized heating was achieved by placing the polymeric part on a heated substrate that utilizes an epoxy matrix. The point heat sources were created by embedding 25 mm long, 1.6 mm diameter brass rods into the epoxy at the two point heat source locations. The top of the rods was flush with the top surface of the epoxy, and the remaining length of the rods protruded below the 4 mm thick epoxy specimen. The temperature in the brass rods was increased by applying resistive heating with constant power. Figure 5.10a shows the temperature field in the heated substrate just before placing the polymer on it, with a maximum temperature $u_{\max} \approx 369$ K. The measured temperature field in the polymer in the surface opposite to the substrate is given in Figure 5.10b. The diffused temperature field in this surface has a maximum value $u_{\max} \approx 322$ K, a 47 K difference with respect to the substrate maximum temperature. A finite element solution of the 3D model is given in Figure 5.10c. The applied heat flux and the heat transfer coefficient are obtained such that the temperature field matches those shown in Figures 5.10a and 5.10b, resulting in $\bar{q} = 23.5\ \mathrm{W/m^2}$ and $h = 1.78\ \mathrm{W/m^2K}$. Thus, it is assumed that the maximum temperature in the polymer is the same as that of the heated substrate. The 3D temperature field given in Figure 5.10c is our reference solution in the absence of flow for the optimization process.

The optimization problem is then stated as:

$$\begin{aligned} &\text{minimize} && \Phi = \left\{\phi_v, \phi_f, \phi_T\right\}, \\ &\text{such that} && \psi_c = 0. \end{aligned}$$

Two approaches are considered to optimize the microvascular network: The first one considers a full 3D optimization, i.e., the thermal boundary value problem is solved using the finite element model used to obtain the temperature field displayed in Figure 5.10c. The second one uses the aid of a 2D optimization by exploiting the two-dimensional nature of the microvascular network template. In other words, a 2D GA optimization can be carried out, and the resulting population from the evolution process can then be used as the starting population for a 3D optimization. Thus, the starting population of the 3D optimization is not random anymore, so particular features about optimal solutions are incorporated into the optimization. This type of optimization belongs to a category

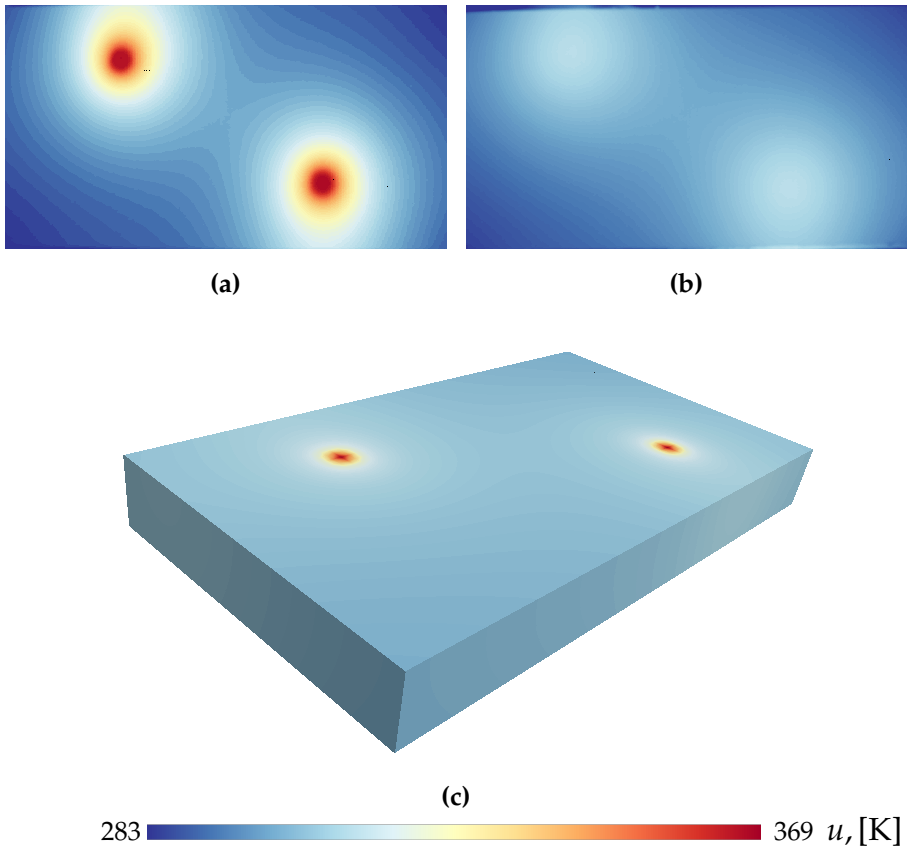


Figure 5.10 – Experimental temperature field obtained by B. Kozola at the University of Illinois by using the setup illustrated in Figure 5.2 for (a) the heated substrate before placing the polymeric component on it; and (b) the polymer surface located opposite to the heated substrate. (c) Temperature distribution from a 3D finite element simulation, obtained by matching the maximum temperatures in Figures (a) and (b) in their corresponding surfaces.

known as knowledge-based GAs.

For the full 3D optimization, a population size $|\mathcal{P}| = 1000$ is used, and the NSGA-II results at generation $t = 5000$ are displayed in Figure 5.11. As before, the front is projected into three orthogonal planes to ease the visualization. The results shown were obtained after running the code for approximately two months on 64 processors. Because the evaluation of the population at a particular generation is proportional to the evaluation time of the individual objective functions, and because of the large finite element mesh used to solve this problem in 3D, the total computational time is dramatically increased with respect to the solution of the previous problem. Three individuals on the front, marked with ◇, ◯, and □ symbols, are displayed in Figure 5.12. The selected individuals show

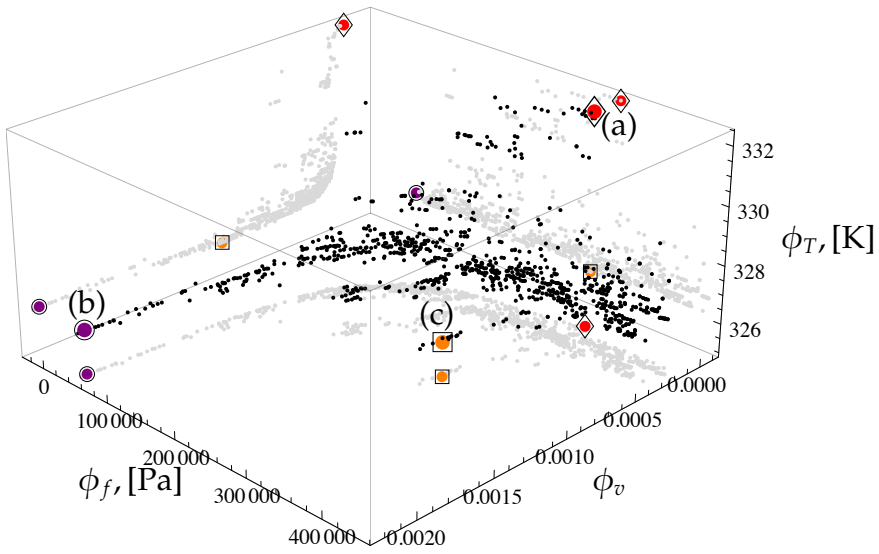


Figure 5.11 – Pareto-optimal surface at generation $t = 5000$ for the localized heating example. The optimization is carried out using the 3D finite element model illustrated in Figure 5.10c. Selected candidate solutions are displayed in Figure 5.12 and are labeled in the front and marked with ◇, ◯, □ symbols.

the temperature field over the surface opposite to the heated substrate, together with their corresponding network and flow field. Arrows in the figure are again scaled according to the magnitude of the flow. The individual shown in Figure 5.12a minimizes the value of ϕ_v in this generation. Clearly, this solution approaches the optimal configuration for this objective function, which connects the inflow and outflow with a straight path. The candidate solution presented in Figure 5.12b maximizes the flow efficiency, and it has a very dense network structure, resembling that of the microvascular network template. Finally, the individual in Figure 5.12c minimizes the maximum temperature objective function ϕ_T. It can be noted that the flow is spread close to the hot spots.

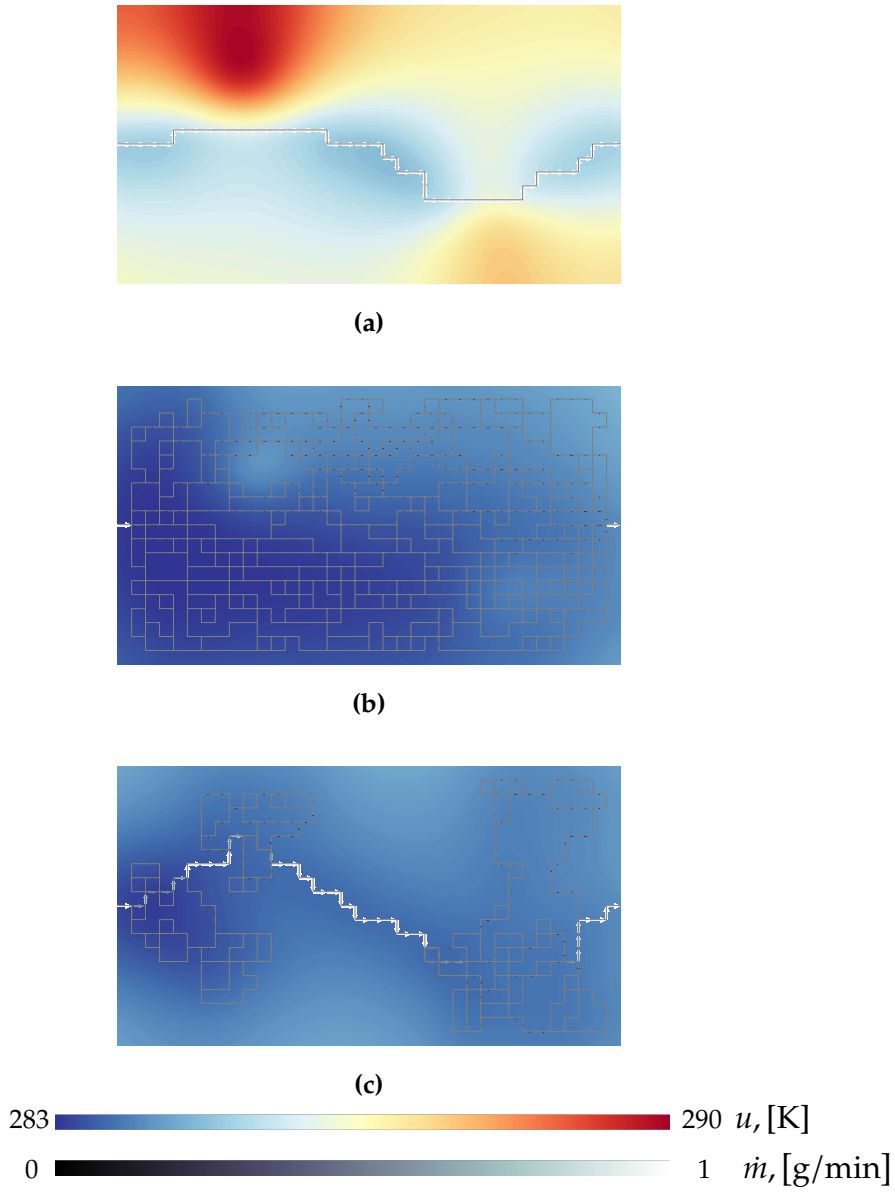


Figure 5.12 – Selected individuals for the localized heating example obtained with the conventional GA scheme after 5000 generations. The figures show the temperature field at the control surface (i.e., the surface opposite to where the localized heat sources are applied) and their corresponding networks and flow fields. The temperature range is defined as the minimum and maximum temperature values in the figures, so they cannot be compared to the control surface temperature field shown in Figure 5.10b. The candidate solutions shown in Figures (a), (b), and (c), minimize objective functions ϕ_v, ϕ_f, and ϕ_T, respectively.

For the second set of results, the thermal boundary value problem is solved using a 2D structured finite element discretization of the main surface, with 5760 three-node elements and 2993 nodes. The applied heat flux is scaled in order to match the maximum temperature field in the heated substrate. The 2D evolution uses a population size $|\mathcal{P}| = 3200$ and the selected individuals at generation $t = 10000$ that minimize objective functions ϕ_v, ϕ_f, and ϕ_T are displayed in Figure 5.13. It took only two days for the algorithm to evolve 10000 generations on a distributed-memory machine with 64 processors. Note the considerable computational time savings from using a 2D solution for the thermal boundary value problem. Note also that for the 2D optimization, the number of generations were doubled and that a much larger population size was used. The individuals that minimize ϕ_v and ϕ_f objective functions, shown in Figures 5.13a and 5.13b, are similar to those shown in Figures 5.12a and 5.12b. However, the candidate shown in Figure 5.13c is remarkably different. The flow is collected into a single path of microchannels that pass through the two hot regions of the domain. The maximum temperature in the 3D computational model considering the network of Figure 5.13c is $u_{\max} = 327.9$ K. On the other hand, the individual shown in Figure 5.12c seems to distribute the flow close to the hot regions, so it would seem that this solution is not the most effective in cooling down the polymeric component. However, the maximum temperature in the polymeric component is $u_{\max} = 326.2$ K, which is 1.7 K below the earlier temperature value.

The 2D population at generation $t = 9900$ is now taken as the starting point for the 3D optimization, and the algorithm is evolved only 100 generations. The candidate that minimizes ϕ_T at generation $t = 10000$ for this knowledge-based evolution is illustrated in Figure 5.14. This solution has a maximum temperature that is only 0.162 higher than the individual in Figure 5.12c. This clearly demonstrates that the network displayed in Figure 5.13c is not the optimal one at minimizing the maximum temperature objective function ϕ_T. Because the heat sources are applied at a plane parallel to that of the microvascular network template, and because of the diffusion of the heat that spreads the hot regions at the plane of the network, the optimal solutions from 2D and 3D cases are remarkably different. However, it is worth noting that a Pareto-optimal front can be obtained faster with the aid of a 2D optimization in these cases.

5.7 Conclusions

This chapter presented a framework for the topology optimization of 3D microvascular flow networks embedded in polymeric materials through the use of GAs. By defining a set

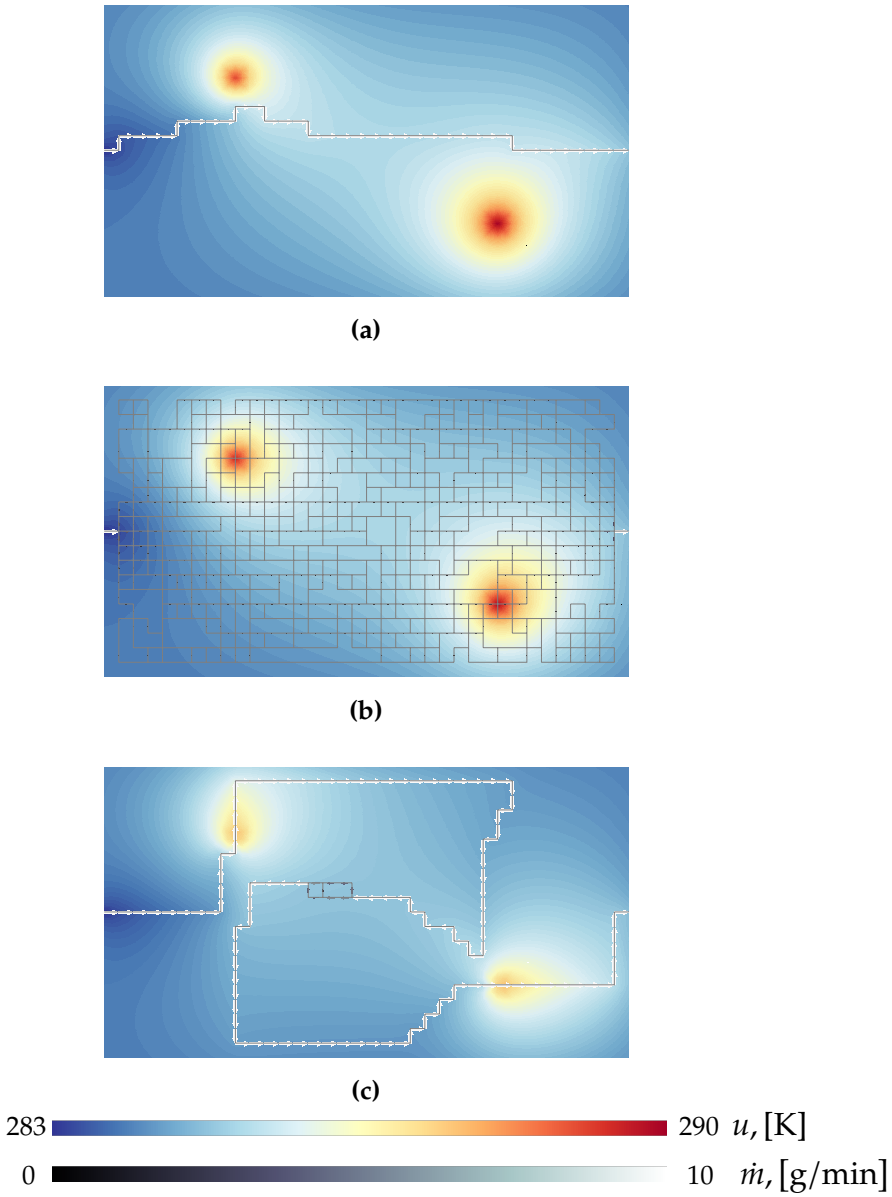


Figure 5.13 – Figures (a), (b), and (c) show the selected individuals at generation t = 10000 of the 2D optimization that minimize ϕ_v, ϕ_f, and ϕ_T objective functions, respectively.

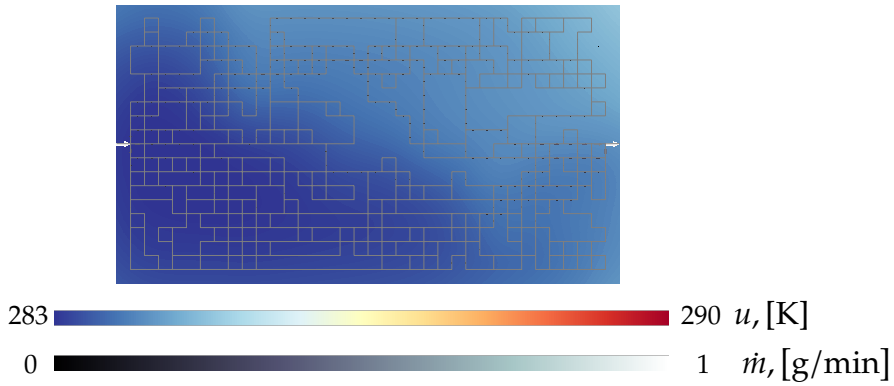


Figure 5.14 – Candidate solution that minimizes the maximum temperature objective function ϕ_T for the knowledge-based GA optimization that started with the evolved population at generation $t = 9900$ of the 2D optimization. The figure shows the temperature field in the surface located away from the application of the heat sources.

of objective functions that describe different physical phenomena, this framework allowed the design microvascular materials for various multi-physics applications. By considering the thermal field in the polymeric component, optimized networks were obtained in the context of active cooling applications. The methodology presented is independent of the number of objective functions and constraints, and can be extended to study other effects, such as maximum strength or maximum stiffness.

The framework was first used to optimize the network topology for a periodic microvascular cell. It was demonstrated that the method was able to capture the trade-offs between conflicting objective functions in a well-defined Pareto-optimal front, from which the selection of a final network design is readily available to the analyst. Periodic boundary conditions were prescribed in the solution of the thermal field to design a cell that was part of larger polymeric thin-walled components.

The second optimization problem, which involves the design of a fin-like microvascular structure subjected to localized heating, was approached using both a conventional GA scheme and a knowledge-based GA, taking advantage of the 2D nature of the embedded network. We demonstrated that seeding the 3D design with the outcome of a 2D optimization yielded optimal network topologies for a fraction of the computational cost. Due to the effect of thermal diffusion, the resulting 3D optimal solutions for the minimization of the maximum temperature were remarkably different from their 2D counterparts.

6 Concluding remarks and future work

Concluding remarks

This work has presented a methodology to carry out the optimization of microvascular networks in polymeric materials. A Genetic Algorithm library was developed in C++ implementing the Simple Genetic Algorithm (SGA) [16], the Deterministic Crowding (DC) algorithm [111], and the Nondominated Sorting Genetic Algorithm II (NSGA-II) [46]. The latter was used in this work as the underlying technique for the optimization of microvascular networks. A constraint space was added to the original NSGA-II in order to handle an arbitrary number of constraints. As a result, the selection mechanism considers the constraint space first, so that unfeasible individuals are discarded over feasible ones. For all optimization problems studied, unfeasible individuals vanished after a few generations of the algorithm. The methodology presented was able to accurately capture the trade-offs between competing objective functions in a Pareto-optimal front. Consequently, the analyst can readily determine the candidate solution that suits a particular problem. No further complexity is introduced by adding objective functions or constraints. Note however, that the visualization of the Pareto-optimal fronts is cumbersome for more than three objective functions, even though an analysis can still be made by visualizing objectives in groups of three.

In Chapter 2, the optimization technique was used for the design of 2D and 3D networks for flow efficiency and void volume fraction objective functions, together with imposed constraints on the networks to ensure adequate redundancy and to eliminate topologies with zero-flow microchannels. The tool was then used to obtain the optimal network topology that distributes the flow evenly to the upper surface of a polymeric component, by considering a flow homogeneity objective function. Chapter 2 also showed that the optimization strategy was completely independent of the dimensionality of the problem, the latter being completely defined by the microvascular network template used as the starting structure. It was showed that a sequential algorithm was able to capture Pareto-optimal fronts of fairly complex 3D optimization problems. However, as the problem increases in size, the initial population of the GA and the number of generations needed

to obtain optimal solutions increases. Hence, it was necessary to take advantage of the embarrassingly parallel nature of genetic algorithms so that the optimization tool scales with the size of the problem. A master-slave approach was developed using the Message Passing Interface (MPI). The parallel algorithm was used for the optimization of the problems discussed in Chapters 4 and 5.

In the context of the physics of the problem, the underlying optimization tool was combined with a finite element solver for the optimization of multi-physics problems involving flow and thermal fields. Consequently, a GFEM library was developed in C++ to solve for the temperature field in 2D and 3D, with Dirichlet, Neumann, and Robin boundary conditions, and any configuration of thermal loads. In Chapter 3, enrichment functions for the GFEM were specially developed to study this problem, where the solution to the thermal field has discontinuous gradients. These enrichment functions were studied in the context of junctions, but they reduce to functions that had been available in the literature for the case of simple interfaces. Even though most of the problems in this work have involved the use of a matching mesh, because of the topology of the microvascular network template, it was demonstrated that optimal convergence is recovered with the use of these enrichment functions when using a non-matching mesh. It was also demonstrated that higher order approximations do not require any correction to the GFEM.

It was shown in Chapter 4 that Pareto-optimal fronts can be easily obtained when a 2D polymeric material is subjected to different loading scenarios. A simplified formulation was used to reduce the computational time needed to obtain the temperature field in the solid. It was demonstrated that by using this formulation, expected optimized network topologies could be obtained. A verification example proved the ability of the GA to obtain microvascular networks that collected the entire flow through the region of the domain where the heat source was applied to reduce the maximum temperature in the polymer. Other examples showed that the GA was able to track the application of localized heat sources and even create networks that can increase the convective heat transfer between the surface of a polymeric component and its surrounding environment. Results were reported for constant inflow, where the mass inflow rate remains constant for the evaluation of the flow problem in all candidate solutions. Results were also provided for constant power, where the mass inflow rate is scaled such that the power remains constant for all evaluations. Differences in the Pareto-optimal fronts for the two scenarios were discussed thoroughly. A detailed analysis of the parallelization scheme adopted in this work predicts that there is an optimal number of processors to achieve maximum speed-up for a particular problem size, after which the speed-up is reduced due to communication time.

Chapter 5 extends the optimization framework presented in Chapter 4 to 3D polymeric components. The main motivation for the chapter was to capture the 3D effects of heat transfer, since a 2D model cannot capture the thermal diffusion through the thickness of the specimen. Also, the convective heat transfer between the polymer and all of its surfaces is a better alternative than using of a temperature-dependent heat source to simulate this effect, as discussed in Chapter 4. A 3D model was also needed to cope with current manufacturing developments. Traditionally, microvascular structures in 3D were created by superimposing 2D layers of the fugitive ink. By using new manufacturing techniques, a 3D microvascular network structure can be printed inside a gel that serves as its support, as explained in Chapter 1. A validation study aimed at closing the gap between experimental measurements and computational predictions. It was shown that the mathematical model used to represent the cooling at microchannels was able to predict cooling effects with the same order of magnitude as those obtained experimentally. However, discrepancies were observed, mainly because of the lack of data for the fluid temperature at inlet locations and because of the non-uniformity of the mass flow rate across microchannels.

It was also shown in Chapter 5 how a material cell that is part of a larger polymeric component could be optimized by applying periodic boundary conditions to the thermal boundary value problem. This was necessary due to the fact that the the optimization of 3D materials requires vast computational resources. It was also demonstrated how the results of a 2D optimization could be used to aid the optimization of a very complex 3D optimization when the microvascular network template is two-dimensional. This methodology is known as a knowledge-based GA, because the initial population for the 3D optimization was not created randomly, but some knowledge about optimal topologies was incorporated into the starting population (the evolved population of a 2D optimization in this case). Finally, it was demonstrated that the 2D results for the maximum temperature objective function were not as optimal as those obtained from the 3D optimization, because of the differences that exist in the temperature field in the 2D optimization and in the plane of the network in the 3D case. In other words, the localized heat sources diffuse over the thickness of the specimen so that the areas of high temperature are more spread in the plane of the network.

Future work

For the problems investigated in this work, reaching the Pareto-optimal fronts by using the NSGA-II algorithm was straightforward, and candidate solutions alternated locations

over the front in subsequent generations. However, the algorithm can exchange optimal solutions in tightly crowded areas for non-optimal solutions because of the crowding distance parameter [47]. Note that this parameter, which is used to better populate the fronts, was devised for the optimization of problems with two objective functions [46]. Furthermore, the algorithm has difficulties finding the extremum candidate solutions on the front. An interesting path for further exploration would be the use of other algorithms. One promising candidate, which was shown to be superior to distribute candidate solutions on the Pareto-optimal front, is a developed version of the Generalized Differential Evolution (GDE) [44]. Another interesting area for further investigation would be the use of hybrid approaches, where the multi-objective evolution strategy is used to search the entire decision space, and another search technique (possibly gradient-based) could be used to obtain local optima. Thus, this technique could overcome the pitfalls of the NSGA-II in tracking extremum candidates over the front.

Regarding the computation of the temperature field through the use of the GFEM, the proposed enrichments are studied in the context of 2D thermal boundary value problems only, thus an interesting area for further research would be the extension of these functions to 3D domains and to elasticity problems. As mentioned previously, all optimization problems investigated in this work involved the use of a simplified formulation to account for the convective heat transfer along the microchannels. However, the simplified formulation relies on a series of assumptions, so further investigation is needed to understand under what conditions the simplified formulation breaks down. By not relying on the simplifying assumptions made on this work, different levels of complexity can be obtained. As an extreme scenario, one could obtain the velocity field in the entire network by solving the Navier–Stokes equations, and then solve for the conjugate heat transfer problem to obtain the temperature field in both solid and fluid domains. This should be done iteratively if the properties of the solid and the fluid change with temperature. Even though the use of a more complex model could not be used in the optimization process because of the additional computational time required, it could be used to verify the candidate solutions obtained with the use of the simplified formulation.

The mathematical model assumes that microchannels have circular cross-sections with a specified diameter, but there is an inherent probabilistic component due to the departure from the theoretical diameter and actual shape of the cross-sections in the microchannels of the resulting manufactured networks. In addition, the accumulation of the fugitive ink at junctions, and its evacuation from the resulting polymeric component aggravate the statistical nature of the final flow field. It would be interesting to study the statistical nature of the flow and to provide a methodology that takes this variability into account.

An immediate extension of this work would consider the transient response when the material is subjected to thermal loads. The time necessary to achieve a steady-state response can be used as an objective function, and networks that differ from those obtained in this work may be created if the new objective function conflicts with those defined in Chapters 4 and 5, i.e., the maximum temperature and thermal convected energy functions. Also, further investigation can be carried out by including other physical phenomena. A foreseeable continuation of this work would consider structural effects, by solving the appropriate elasticity boundary value problem. In this way, the loss of stiffness and strength because of the presence of the microvascular network could be quantified, and optimal network geometries that mitigate this problem could be obtained. Also, the stress concentrations due to the presence of the network can be quantified under a variety of load conditions.

It is worth mentioning that the methodology presented in this work can be applied to a large set of problems. The tools developed have been encapsulated in portable libraries that can be installed in most scientific computing environments. Virtually any problem for which an optimal solution cannot be obtained in polynomial time can make use of the `xga` genetic algorithm library. The `yafeq` library provides a flexible implementation for obtaining approximate solutions to boundary value problems through the GFEM. Details on the implementation of these tools are provided in Appendix A.

A Implementation

The implementation details behind the tools used to carry out the multi-physics optimization of microvascular materials are described henceforth. For this project, about 50000 lines of code were written in C++, a strongly typed programming language. The code is divided in four different projects: *i)* `cputils`, *ii)* `xga`, *iii)* `yafeq`, and *iv)* μ`vac`. The first three projects are C++ libraries that are used by the μ`vac` project to carry out the actual optimization of microvascular networks. The four projects make use of the GNU's Not Unix! (GNU) build system, a.k.a. the *autotools*, to handle the configuration of each project depending on the platform where it will be used, for the creation of makefiles and for the installation of static and shared libraries. Each project has been thoroughly documented by using the *Doxygen* project. The following sections describe each of the projects individually.

A.1 The `cputils` library

As the name suggests, `cputils` is a library of C++ utilities that are used by all projects.

A.1.1 `Vector` and `Matrix` classes

C++ does not contain vector and matrix objects for use out of the box, so the developer usually uses objects provided by libraries like the Matrix Template Library (MTL) [112] or the blitz++ library [113]. For this work, new vector and matrix objects were developed in order to reduce the code dependencies. The main motivation was the use of the BLAS library [114, 115], a set of functions for vector and matrix operations that are usually optimized for the scientific computing environment they operate on by taking advantage of the memory hierarchy. A requirement of the use of the BLAS library, is that all elements in vectors and matrices be contiguous in memory. As a result, the vector and class classes developed are wrappers on standard C arrays.

The vector class also defines convenient functions inherent to the abstract concept of a mathematical vector v. Thus, the implementation defines convenient functions to obtain

the size $|v|$ and the norms $\|v\|_1$, $\|v\|_2$ and $\|v\|_\infty$ of a vector, to normalize a vector (such that $\|v\| = 1$), to compute the sum of the elements $\sum_i v_i$, and to obtain maximum and minimum values (i.e., $\max_i v_i$ and $\min_i v_i$). Element access can be achieved by the use of parenthesis (i.e., $v(i)$ gives the $i-$th element) as in Fortran code, or as in standard C arrays (i.e., $v[i]$ gives the $i-$th element). Elements in the vector container can also be accessed by the use of iterators, as in the C++ standard library [116, 117] containers.

Similarly, the matrix class is a wrapper on a one-dimensional C array, where the elements of the two-dimensional array are flattened in column-major order (i.e., storage in the array is done column by column such that every element of the $i-$th column has an index in the array that is lower than any element in subsequent columns). Again, the matrix class provides functions inherent to an abstract mathematical matrix concept. For a matrix M, functions are provided to obtain the number of rows m and the number of columns n, to obtain the norms $\|M\|_1$ and $\|M\|_\infty$, among others. Element access to the elements in the matrix can again be done by the use of parenthesis and brackets, $M(i, j) = M[i][j]$. In C++, `operator[]` is a unary operator, so the element access through the use of brackets is carried out by the creation of a local proxy class [118].

A.1.2 `cppblas`

The vector and matrix classes described in the previous section can be used with the BLAS library because all elements are located in contiguous memory. However, the use of the BLAS routines is not intuitive and it is prone to error. Listing A.1 shows the declaration of the function of the C interface to the BLAS library that carries out the multiplication between two matrices.

```
void cblas_dgemm(const enum CBLAS_ORDER Order,
                 const enum CBLAS_TRANSPOSE TransA,
                 const enum CBLAS_TRANSPOSE TransB, const int M, const int N,
                 const int K, const double alpha, const double *A,
                 const int lda, const double *B, const int ldb,
                 const double beta, double *C, const int ldc);
```

Listing A.1 – The `cblas_dgemm` function for matrix-matrix multiplication, taken from the C interface to the BLAS library

As a result, an interface to the functions provided by the BLAS library was written to support the use of mathematical operators. Thus, when the client types A * B for the multiplication between matrices A and B, the interface calls the `cblas_dgemm` function given in Listing A.1. A comprehensive facility was built on the idea behind *expression*

templates [119], the underlying framework behind the blitz++ library [113]. In this way, when the client writes A ∗ B, an expression object is created containing the necessary information to be carried out (i.e., references to the matrices A and B, and the multiplication operation between them). This concept is an example of *lazy evaluation*, a technique used to defer the computation to the time when the result is actually needed. Using expression templates allows the operation 2.0 ∗ (A ∗ (3.0 ∗ B)) to perform a single matrix-matrix multiplication. This is because the compiler creates an expression object for (3.0 ∗ B), which is not evaluated immediately. Another expression object is created as the multiplication of A and (3.0 ∗ B), but the expression is still not evaluated. Finally, another expression is created by the multiplication between 2.0 and the last expression created, updating the value of the scalar used in the multiplication. When the final expression is evaluated, a unique evaluation is carried out, as if the client code written were C ∗ B, with C = 6.0 ∗ A. Furthermore, depending on the entities involved in the expression, the return type of an entire expression can be determined by a metaprogram at compilation time. In other words, with a vector v and matrix M, the result of the mathematical expressions $v'v$, Mv, and vv' is a scalar, a vector, and a matrix, respectively[1]. No matter how complicated an expression is, the template metaprogram can determine the return type of the expression at compilation time.

The code overloads operators +, −, and ∗, which means that these operators are defined for vectors, matrices, and for many expression template objects. In addition, the code supports transposition, matrix inversion, Kronecker product, Gaussian elimination and Cholesky factorization.

A.1.3 The `sparse_matrix` class

The evaluation of the objective functions in this work involves the solution of a system of linear equations with a sparse matrix. For this work, a class template was written such that the elements are stored in hashed associative containers. A hashed associative container (known as a hash map) is a data structure where the value associated with a key is obtained through the use of a hash function. When the keys are such that a simple hash function can be used, the lookup of a value has close to constant time complexity. Thus, these containers reduce considerably the amount of memory needed to store a sparse matrix, compared to the use of a dense matrix implementation (as explained in Section A.1.1), while not deteriorating the performance of element lookup. The `sparse_matrix` class supports the element access through the use of `operator()` and `operator[]`, similarly to the dense

[1]The expression v' denotes the transpose of a vector, i.e., a row vector.

matrix of Section A.1.1. In addition, operations between sparse matrices is very efficient because they operate on non-zero entries only. In addition, the `sparse_matrix` defines local classes `CCS` and `CRS`, which can be used to convert the matrix to compressed-column and compressed-row storage formats, respectively. The compressed-column storage is also known as the Harwell-Boeing sparse matrix format [120]. In this way, the sparse matrix can be represented by the use of three arrays, two of which can be used for element access, and the last one to store the non-zero values of the matrix. For a sparse matrix of size $n \times n$, any of these formats require $2z + n + 1$ storage locations, with z being the number of non-zero entries in the sparse matrix, which is considerably better than the locations needed for a dense matrix n^2. The use of these formats is required by the `superlu` library [121], which is used for the computation of the solution vector when the matrix is sparse.

A.1.4 Solvers

Two solvers provide the bridge between the data structures defined in the previous sections and the `superlu` library [121]. The first solver suits most problems and is a wrapper on the C routines implemented by the `superlu` library. The second solver is used when the matrix is semi-definite, so it implements the algorithm described in [58, 83], and briefly outlined in Algorithm 3. In this algorithm, the solution vector is obtained by modifying the matrix such that it is ill-conditioned but not singular. Then, the solution is obtained by carrying out iterative refinement on this initial vector.

Algorithm 3 Positive Semi–Definite Solver algorithm

procedure SEMIDEFINITESOLVER(K, u, ϵ)
  for $i = 1$ to n **do**
    $T_{ii} = 1/\sqrt{K_{ii}}$ ▷ *builds transformation matrix*
  end for
  $\bar{\mathbf{K}} = \mathbf{TKT}$ ▷ *applies transformation to* $\mathbf{K}$
  $\bar{\mathbf{f}} = \mathbf{T}\,\mathbf{f}$ ▷ *applies transformation to* $\mathbf{f}$
  $\bar{\mathbf{K}}_\epsilon = \bar{\mathbf{K}} + \epsilon\mathbf{I}$ ▷ *computes perturbed matrix*
  Solve $\bar{\mathbf{K}}_\epsilon \bar{\mathbf{u}}_0 = \bar{\mathbf{f}}$ for $\bar{\mathbf{u}}_0$ ▷ *computes initial solution* $\mathbf{u}_0$
  $\mathbf{r}_0 = \bar{\mathbf{f}} - \bar{\mathbf{K}}\bar{\mathbf{u}}$ ▷ *computes residual* $\mathbf{r}_0$
  while $\varepsilon > tol$ **do** ▷ *start iterative refinement*
    Solve $\bar{\mathbf{K}}_\epsilon \mathbf{e} = \mathbf{r}$ for $\mathbf{e}$
    $\mathbf{r} = \bar{\mathbf{f}} - \bar{\mathbf{K}}\bar{\mathbf{u}}$
    $\bar{\mathbf{u}} = \bar{\mathbf{u}} + \mathbf{e}$
    $\varepsilon = \mathbf{e}^T\bar{\mathbf{K}}\mathbf{e}/\bar{\mathbf{u}}^T\bar{\mathbf{K}}\bar{\mathbf{u}}$
  end while
end procedure

A.1.5 The `ordered_tree` class template

An ordered tree is a data structure where the children of every node respect and maintain an ordering. It was necessary to write such data structure because the C++ standard library [116] does not provide such functionality. The `ordered_tree` class template was developed from the red-black tree implementation found in the C++ standard library. The data structure provides several types of iterators, depending on the order of traversing the nodes of the tree. Thus, the class provides depth-first iterators, breadth-first iterators, leaf iterators, and even children iterators. The data structure supports functions to count elements of certain value, find, insert and erase values, among others.

A.1.6 Other functionality

The library implements a set of functions for pseudo-random number generations. These include a function to set the seed, a function that returns a value between 0 and 1, or between two values $x_{\min}$ and $x_{\max}$, a function that simulates the flip of a biased coin (i.e., a Bernoulli trial), and a function that returns a normally distributed value with zero mean and unit variance. The library also implements a `timer` class, an expression calculator, used for get time measures in the code. It implements a `tokenizator` class, used to divide a string into tokens, macros for static assertions [122], geometry classes defining the abstract concepts of points, segments, and bounding boxes, and a non-strict implementation of the *visitor design pattern* [123, 122].

A.2 The `xga` library

The `xga` library, short for extended genetic algorithms, implements the following algorithms: *i)* the Simple Genetic Algorithm (SGA) [16] , *ii)* the Deterministic Crowding (DC) algorithm [111], and *iii)* Nondominated Sorting Genetic Algorithm II (NSGA-II) [46]. The former two algorithms deal with a single objective function, whereas the latter is the multi-objective and constrained algorithm chosen in this work to carry out the optimization of the microvascular networks. The library was written on Andrei Alexandrescu's *policy-based design* concept [122]. By combining the template mechanism of C++ with multiple inheritance, policy-based design is a powerful methodology to confront designs that are exponential in nature. With a small set of classes, which are called *policy classes* in this context and that are used as template parameters to other classes, the library can define a

large set of genetic algorithms by assembling them analogously to a Russian matryoshka doll. Another important property of using this approach is that every single aspect of the resulting implementation is determined at compilation time, thereby allowing the use of compile-time assertions [122].

The library implements class templates for the abstract concepts of a chromosome, an individual, a population, for the genetic operators and for the genetic algorithms. The basic building block of the code is the `Objective` class template, which contains information about the number of objective functions and constraints used, and a parameter for each one of them that indicates if maximization or minimization is carried out. In this way, it is possible to minimize some objective functions while maximizing others. The next building block is the `Chromosome` class template, which takes the `Objective` class template as a building block. The chromosome contains a structure that store its alleles, and information about the type of mutation carried being carried out. The `Chromosome` class is itself a building block for the `Individual` class template, which embeds the structure of the former. The `Population` class template then takes the `Individual` class as a building block to configure the data structures used to store individual objects. The `Population` class template contains information of how individuals crossed. Finally, a specific class of genetic algorithm, taken from those mentioned in the first paragraph of this section, takes the `Population` class as a building block, and also contains information about the selection mechanism. The concept of genetic operators is therefore implemented in three categories of policy classes. The mutation policy classes are passed as a building block to the `Chromosome` class template. Mutation policy classes implemented include a bitwise mutation policy, a clock bitwise mutation policy, and a mutation policy used for integers that uses a gaussian distribution function. A crossover policy class is used as a building block for the `Population` class template. Crossover policy classes implemented carry out one-point, and uniform crossover. Finally, the selection policy classes are used to configure the genetic algorithm class template, and only a selection policy that implements tournament selection is available at this time. The library can be easily extended to include other policy classes implementing further selection, crossover and mutation mechanisms, as long as the new policy classes respect the interface of the policy.

The library can be configured to be used in distributed memory machines. An implementation of the master-slave approach was developed using the Message Passing Interface (MPI), a commonly used protocol for parallel computation. As mentioned in Chapters 2 and 4, the evaluation part of the genetic algorithm is embarrassingly parallel. Therefore, the master slave sends the information about the chromosomes in the population to several processes. The processes evaluate the chromosomes that they receive and

send the results back to the master. A thorough discussion on the methodology adopted can be found in Chapter 4.

A.3 `yafeq` (yet another finite element qode)

The functionality of the `xga` library is fully determined at compilation time. On the other hand, the behavior of the `yafeq` library is determined at runtime. This is because all the information to run a simulation needs to be read from a file, so there are less opportunities for static assertions. The library reads standard Abaqus files and even extends its semantics to provide functionality that is not available. `yafeq` can solve thermal and elasticity boundary value problems in two and three dimensions. Most major class templates in the library inherit from their corresponding abstract base classes. To provide the concrete functionality in derived classes, functions are overridden, the typical approach that polymorphism provides in object oriented programming. However, in this work, there is a vast use of the visitor design pattern [123, 122]. In this way, classes that support the visitor design pattern implement a bouncing function and the further functionality can be encapsulated in the visitor class definition, without adding virtual functions. The suggested approach when using visitor is to use *strict visitation*, which means that the compiler prevents the compilation if the behavior is not implemented to at least one class in the visiting hierarchy. However, there are functions that make sense for some classes and not for others, even in the same category. Thus, a non-strict implementation of the visitor design pattern is used, which can be found as one of the components of the `cpputils` library.

The objects in the library model the abstract concepts found in the finite element method, i.e., nodes, elements, element sides, materials, boundary conditions, equations, meshes and the finite element itself. Because the information about the creation of these objects is not known at compilation time, the creation is done through the use of object factories contained in singleton objects (see [123] and [122] for an explanation on these design patterns). Additionally, the library implements a set of classes to support the generalized finite element method, by adding enrichment classes. Most classes in the code provide iterators to access the state stored in the classes, similarly to the containers in the C++ standard library [116, 117]. Most of the class templates in the library contain the dimensionality of the problem as a template parameter, which is very useful when implementing classes that have different behavior in two and three dimensions.

The base node class contains information about its location on space and elements it is

connected to (i.e., the node connectivity table). A mesh node inherits this information and stores information about the degrees of freedom and solution values for the equations that are solved. For the element classes, the hierarchy contains an abstract base class, a middle class that implements common functionality to all elements using the Curiously Recurring Template Pattern (CRTP) [124], and several concrete classes that implement the different finite elements supported by the library. The element base class embeds a container of side objects, contains the node connectivity table, and the material the element is made of. For 2D problems, the library currently implements linear, quadratic and cubic triangles, and bilinear quadrangles. 3D elements include tetrahedra, hexahedra (brick elements) and triangular prism elements (i.e., wedge elements). The material base class store information about material properties, such as material density, specific heat, conductivity, elastic moduli, Poisson's ratio, etc. Concrete classes define the material behavior as elastic, elastic under plane-strain (or plane-stress), plastic, etc. The material model dictates the assembly of the element local stiffness matrices. Boundary conditions classes implement Dirichlet, Neumann, Robin, and periodic boundary conditions. Equation objects store information about the number of degrees of freedom per node for the solution of its corresponding boundary value problem, and embed a container of boundary conditions. Concrete equation objects include a thermal equation and a structural equation (for elasticity problems). A mesh class embeds containers nodes, elements, materials and equation objects. As a result, it is possible to solve two different equations for the same mesh. Note again that the boundary conditions for a boundary value problem are associated with the equation objects. It also contains the data structures that store the stiffness matrices and vectors. Elements in the mesh are stored in an ordered tree container, explained in Section A.1.5. In the context of the GFEM, when an element is subdivided into integration elements (see Chapter 3), the latter are added as children of the former in the ordered tree hierarchy. As a result, the assembly of finite elements needs only traverse the elements found as leaves of the ordered tree. The solution of the system of equations is carried out by the solvers described in Section A.1.4, and the resulting field is stored in the nodes of the mesh. The mesh also provides a function for the evaluation of a field at a point, functionality that is carried out by the creation of a quad-tree in 2D (or an oct-tree in 3D), which allows to find the element that contains the point in logarithmic time complexity. It also implements functions for obtaining maximum and minimum field values, computing error measures (see Equations (3.4.1) and (3.4.1)), energy values, and to export the mesh so that it can be visualized by the Visualization Toolkit (VTK). Finally, a finite element class template embeds a container of meshes, and provides the user with the interface for I/O. The template parameters of this class control the type of formulation used (either standard FEM

or GFEM) and the type of analysis carried out (steady or transient).

As mentioned above, other classes were developed to support the GFEM. The use of the GFEM allows us to obtain accurate solutions in finite element meshes that are completely independent of the geometry of the network, as shown in Figure A.1. The enrichment functions presented in Chapter 3 were developed because of the fact that the gradient of the temperature field in the direction perpendicular to a microchannel is discontinuous, as displayed in Figure A.2. As explained in Chapter 3, the generation of matching meshes is not straightforward when the topology of the microvascular network is complicated. For the GFEM, a special class takes care of enriching the nodes of the

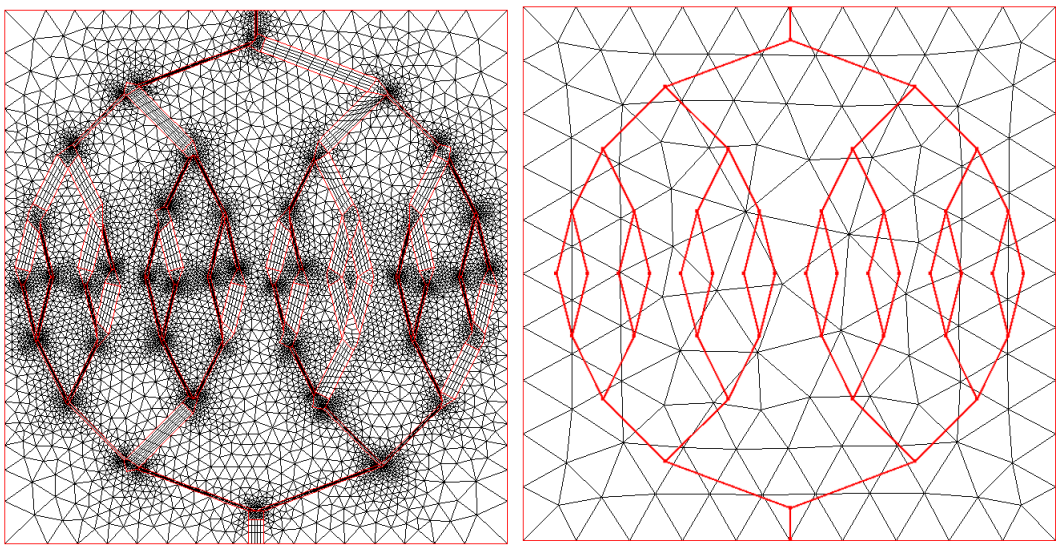


Figure A.1 – Finite element meshes: a) Standard FEM; b) GFEM.

finite elements that are intersected by the network, increasing the number of degrees of freedom and assigning enrichment functions for the computation of shape functions. The implementation supports all the enrichment functions described in Chapter 3. Finite elements that are intersected by the microvascular network are split into integration elements. In 2D, the algorithm used is described in Algorithm 4 In the algorithm, several special cases are taken into account. For a three-node triangle, these cases are illustrated in Figure A.3.

Chapter 3 also explains the use of adaptive integration to find regions of the mesh that require special attention due to use of the enrichment functions. The approach adopted in that chapter is outlined in Algorithm 5, which makes use of the function outlined in Algorithm 4. A parent element at the root is recursively split into integration elements

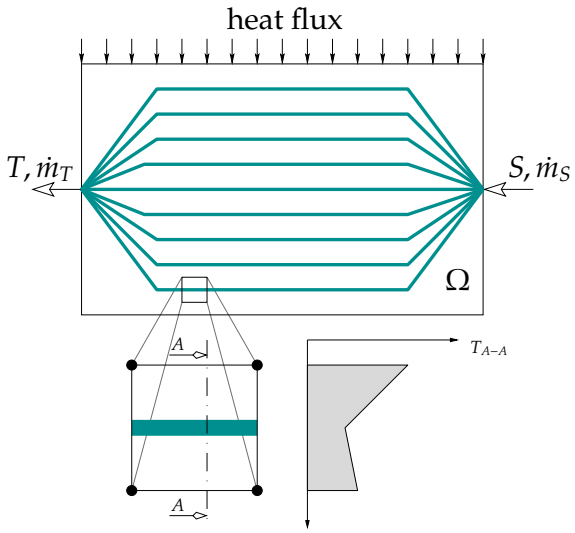


Figure A.2 – Schematic of the temperature field close to a microchannel.

Algorithm 4 Algorithm used to split 2D finite elements into integration elements.

```
procedure Split(M)
    for all N_i ∈ M do                                  ▷ loop over nodes
        φ_i ← LevelSetFunction(N_i)                     ▷ get level set function
    end for
    for all E_i ∈ M do                                  ▷ loop over elements
        {c} ← ∅                                         ▷ initialize point container
        for j ← 1,Edges(E_i) do                         ▷ loop over element edges
            if φ_s · φ_t < 0 then
                {c} ← {c} ∪ IntersectionPoint()         ▷ add point to
container
            else if φ_s = 0 or φ_t = 0 then
                SpecialCase()                           ▷ handle special case
            else
                do nothing
            end if
        end for
        if |{c}| ≥ 2 then
            Cut(E_i, {c})                               ▷ cut element
        end if
    end for
end procedure
```

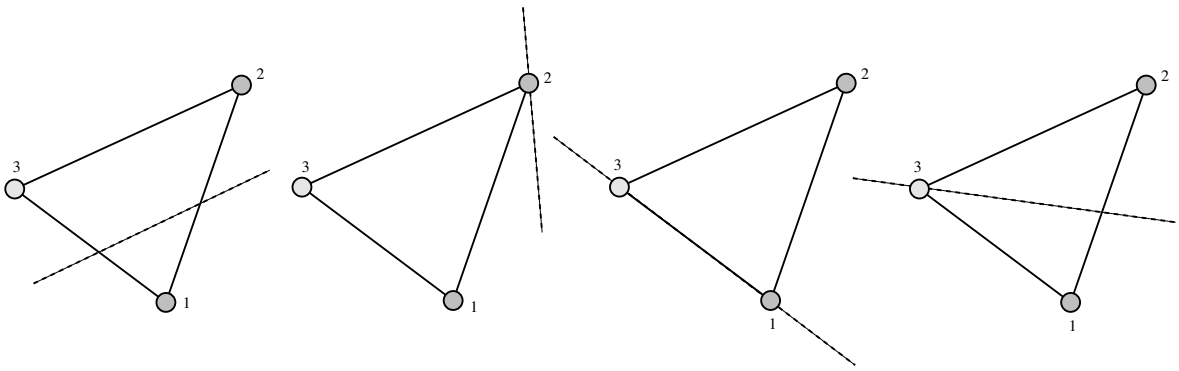
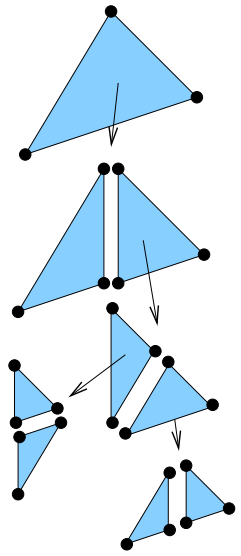


Figure A.3 – Special cases for handling the cut of a three-node triangle depending on the value of the level set function ϕ. From left to right: *i)* $\phi_1 < 0$, $\phi_2 > 0$, $\phi_3 > 0$; *ii)* $\phi_1 < 0$, $\phi_2 = 0$, $\phi_3 < 0$; *iii)* $\phi_1 = \phi_3 = 0$, $\phi_2 > 0$; and *iv)* $\phi_1 < 0$, $\phi_2 > 0$, $\phi_3 = 0$.

until errors in the norm of matrices and vectors with respect to the previous recursion lie within the specified tolerance.

Algorithm 5 Adaptive integration

procedure INTEGRATE(p, K, F)
  $\{k_p, f_p\} \leftarrow$ ASSEMBLE(p)
  $\{c_1, c_2\} \leftarrow$ SPLIT(p)
  $\{k_{c_1}, f_{c_1}\} \leftarrow$ ASSEMBLE($c1$)
  $\{k_{c_2}, f_{c_2}\} \leftarrow$ ASSEMBLE($c2$)
  $r_K \leftarrow \left\| k_p - (k_{c_1} + k_{c_2}) \right\|$
  $r_F \leftarrow \left\| f_p - (f_{c_1} + f_{c_2}) \right\|$
  if $r_K > tol$ **or** $r_F > tol$ **then**
    INTEGRATE(c_1)
    INTEGRATE(c_2)
  else
    $K \leftarrow K + k_p$
    $K \leftarrow F + f_p$
  end if
end procedure



A.4 The μvac project

The μvac project makes use of the tools described in the previous sections to carry out the optimization of the topology of microvascular networks in 2D and 3D. The microvascular network is implemented as a graph data structure using the Boost Graph Library (BGL) [125], which allows a flexible implementation with custom properties associated to vertices

and edges. The BGL provides also convenient algorithms for the traversal of vertices and edges, and the algorithm for the computation of the number of connected components of the graph, widely used to find unfeasible networks in this work. The μvac also implements the interface for the computation of objective functions.

References

[1] Y. Bar-Cohen. Biomimetics – using nature to inspire human innovation. *Bioinspiration & Biomimetics*, 2006; 1(1): P1. URL http://dx.doi.org/10.1088/1748-3182/1/1/P01.

[2] P. Fratzl. Biomimetic materials research: what can we really learn from nature's structural materials? *Journal of The Royal Society Interface*, 2007; 4(15): 637–642. URL http://dx.doi.org/10.1098/rsif.2007.0218.

[3] L. White. Eilmer of Malmesbury, an Eleventh Century Aviator. *Technology and Culture*, 1961; 2(2): 97–111.

[4] C. Pedretti and C. Frost. *Leonardo: art and science*. Firenze: Giunti, 2000. ISBN 8809015118.

[5] J. F. V. Vincent. Biomimetics – a review. *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, 2009; 223.

[6] B. Bhushan. Biomimetics: lessons from nature - an overview. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2009; 367(1893): 1445–1486. URL http://dx.doi.org/10.1098/rsta.2009.0011.

[7] S. R. White, N. R. Sottos, P. H. Geubelle, J. S. Moore, M. R. Kessler, S. R. Sriram, E. N. Brown, and S. Viswanathan. Autonomic healing of polymer composites. *Nature*, 2001; 409(6822): 794–797. URL http://dx.doi.org/10.1038/35057232.

[8] K. S. Toohey, N. R. Sottos, J. A. Lewis, J. S. Moore, and S. R. White. Self-healing materials with microvascular networks. *Nature Materials*, 2007; 6(8): 581–585. URL http://dx.doi.org/10.1038/nmat1934.

[9] C. Hansen, W. Wu, K. Toohey, N. Sottos, S. White, and J. Lewis. Self-Healing Materials with Interpenetrating Microvascular Networks. *Advanced Materials*, 2009; 21(41): 4143–4147. URL http://dx.doi.org/10.1002/adma.200900588.

[10] D. Therriault, S. R. White, and J. A. Lewis. Chaotic mixing in three-dimensional microvascular networks fabricated by direct-write assembly. *Nature Materials*, 2003; 2(4): 265–271. URL http://dx.doi.org/10.1038/nmat863.

[11] A. Bejan and S. Lorente. Constructal theory of generation of configuration in nature and engineering. *Journal of Applied Physics*, 2006; 100(4): 041301. URL http://dx.doi.org/10.1063/1.2221896.

[12] A. Bejan, S. Lorente, and K.-M. Wang. Networks of channels for self-healing composite materials. *Journal of Applied Physics*, 2006; 100(3): 033528. URL http://dx.doi.org/10.1063/1.2218768.

[13] H. Zhang, S. Lorente, and A. Bejan. Vascularization with trees that alternate with upside-down trees. *Journal of Applied Physics*, 2007; 101(9): 094904. URL http://dx.doi.org/10.1063/1.2723186.

[14] S. Kim, S. Lorente, and A. Bejan. Vascularized materials: Tree-shaped flow architectures matched canopy to canopy. *Journal of Applied Physics*, 2006; 100(6): 063525. URL http://dx.doi.org/10.1063/1.2349479.

[15] A. Klarbring, J. Petersson, B. Torstenfelt, and M. Karlsson. Topology optimization of flow networks. *Computer Methods in Applied Mechanics and Engineering*, 2003; 192(35-36): 3909–3932. URL http://dx.doi.org/10.1016/S0045-7825(03)00393-1.

[16] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Massachusetts, 1989. ISBN 0201157675.

[17] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Massachusetts, 2002. ISBN 1402070985.

[18] A. M. Aragón, J. K. Wayer, P. H. Geubelle, D. E. Goldberg, and S. R. White. Design of microvascular flow networks using multi-objective genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 2008; 197(49-50): 4399–4410. URL http://dx.doi.org/10.1016/j.cma.2008.05.025.

[19] A. M. Aragón, C. A. Duarte, and P. H. Geubelle. Generalized finite element enrichment functions for discontinuous gradient fields. *International Journal for Numerical Methods in Engineering*, 2010; 82(2): 242–268. URL http://dx.doi.org/10.1002/nme.2772.

[20] G. Williams, R. Trask, and I. Bond. A self-healing carbon fibre reinforced polymer for aerospace applications. *Composites Part A: Applied Science and Manufacturing*, 2007; 38(6): 1525 – 1532. ISSN 1359-835X. URL http://dx.doi.org/10.1016/j.compositesa.2007.01.013.

[21] J. W. Pang and I. P. Bond. A hollow fibre reinforced polymer composite encompassing self-healing and enhanced damage visibility. *Composites Science and Technology*, 2005; 65(11-12): 1791 – 1799. ISSN 0266-3538. URL http://dx.doi.org/10.1016/j.compscitech.2005.03.008.

[22] L. A. Shipton. *Thermal management applications for microvascular systems*. Master's thesis, University of Illinois at Urbana-Champaign, 2007.

[23] W. Wu, C. J. Hansen, A. M. Aragón, N. R. Sottos, S. R. White, P. H. Geubelle, and J. A. Lewis. Direct ink writing of microvascular networks. In *Proceedings of the First International Conference on Self Healing Materials*. Noordwijk aan Zee, The Netherlands, 2007; .

[24] T. Borrvall, A. Klarbring, J. Petersson, and B. Torstenfelt. Topology Optimization in Fluid Mechanics. In *Fifth World Congress on Computational Mechanics*. Vienna, Austria, 2002; .

[25] A. R. Simpson, G. C. Dandy, and L. J. Murphy. Genetic Algorithms Compared to Other Techniques for Pipe Optimization. *Journal of Water Resources Planning and Management*, 1994; 120(4): 423–443. URL http://dx.doi.org/10.1061/(ASCE)0733-9496(1994)120:4(423).

[26] K. Vairavamoorthy and M. Ali. Optimal Design of Water Distribution Systems Using Genetic Algorithms. *Computer-Aided Civil and Infrastructure Engineering*, 2000; 15(5): 374–382. ISSN 10939687. URL http://dx.doi.org/10.1111/0885-9507.00201.

[27] T. D. Prasad and N.-S. Park. Multiobjective Genetic Algorithms for Design of Water Distribution Networks. *Journal of Water Resources Planning and Management*, 2004; 130(1): 73–82. URL http://dx.doi.org/10.1061/(ASCE)0733-9496(2004)130:1(73).

[28] S. V. Kumar, T. A. Doby, J. John W. Baugh, E. D. Brill, and S. R. Ranjithan. Optimal Design of Redundant Water Distribution Networks Using a Cluster of Workstations. *Journal of Water Resources Planning and Management*, 2006; 132(5): 374–384. URL http://dx.doi.org/10.1061/(ASCE)0733-9496(2006)132:5(374).

[29] L. Davis (editor). *Handbook of genetic algorithms*. Van Nostrand Reinhold Company, 1st. edition, 1991. ISBN 0442001738.

[30] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd. edition, 2000. ISBN 0130144002.

[31] A. M. Aragón, C. J. Hansen, W. Wu, P. H. Geubelle, J. Lewis, and S. R. White. Computational design and optimization of a biomimetic self-healing/cooling composite material. In *Proceedings of SPIE*, volume 6526. SPIE, San Diego, California, USA, 2007; 65261G. URL http://dx.doi.org/10.1117/12.717064.

[32] C. A. Brebbia and A. J. Ferrante. *Computational Hydraulics*. Butterworth-Heinemann Ltd, London, 1983. ISBN 040801153X.

[33] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications, 1998. ISBN 0486402584.

[34] K. Steiglitz, P. Weiner, and D. Kleitman. The Design of Minimum-Cost Survivable Networks. *IEEE Transactions on Circuits and Systems*, 1969; 16(4): 455–460. ISSN 0018-9324. URL http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1083004&isnumber=23404.

[35] J. Edmonds and R. M. Karp. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *Journal of the Association for Computing Machinery*, 1972; 19(2): 248–264. ISSN 0004-5411. URL http://dx.doi.org/10.1145/321694.321699.

[36] J. H. Holland. Outline for a Logical Theory of Adaptive Systems. *Journal of the Association for Computing Machinery*, 1962; 9(3): 297–314. ISSN 0004-5411. URL http://dx.doi.org/10.1145/321127.321128.

[37] J. H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor, University of Michigan Press, 1975. ISBN 0472084607.

[38] T. Bäck and H.-P. Schwefel. *Evolution Strategies I: Variants and their Computational Implementation.*, chapter 6, Genetic algorithms in engineering and computer science, 111–126. John Wiley & Sons, Inc. ISBN 047195859X, 1995; .

[39] H.-P. Schwefel and T. Bäck. *Evolution Strategies II: Theoretical Aspects.*, chapter 7, Genetic algorithms in engineering and computer science, 127–140. John Wiley & Sons, Inc. ISBN 047195859X, 1995; .

[40] J. T. Richardson, M. R. Palmer, G. E. Liepins, and M. Hilliard. Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the third international conference on Genetic algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN 1-55860-006-3, 1989; 191–197.

[41] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *IEEE World Congress on Computational Intelligence. Proceedings of the First IEEE Conference on Evolutionary Computation*, volume 1. 1994; 82–87.

[42] N. Srinivas and K. Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 1995; 2(3): 221–248. URL http://dx.doi.org/10.1162/evco.1994.2.3.221.

[43] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference (S. Forrest, ed.)*. San Mateo, CA: Morgan Kaufmann, 1993; 416–423.

[44] S. Kukkonen and J. Lampinen. GDE3: The third Evolution Step of Generalized Differential Evolution. In *The 2005 IEEE Congress on Evolutionary Computation*, volume 1. Edinburgh, Scotland. ISBN 0780393635, 2005; 443–450. URL http://dx.doi.org/10.1109/CEC.2005.1554717.

[45] V. Pareto. *Manuale di Economia Politica*. Piccola Biblioteca Scientifica, Milan, 1906. Translated into English by Ann S. Schwier (1971), Manual of Political Economy, MacMillan, London.

[46] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 2002; 6(2): 182 –197. ISSN 1089-778X. URL http://dx.doi.org/10.1109/4235.996017.

[47] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., 1st. edition, 2001. ISBN 047187339X.

[48] M. Pelikan, K. Sastry, and D. E. Goldberg. Multiobjective hBOA, clustering, and scalability. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, New York, NY, USA. ISBN 1-59593-010-8, 2005; 663–670. URL http://dx.doi.org/10.1145/1068009.1068122.

[49] E. Cantú-Paz and D. E. Goldberg. Efficient parallel genetic algorithms: theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 2000; 186(2-4): 221–238. URL http://dx.doi.org/10.1016/S0045-7825(99)00385-0.

[50] N. Sukumar, D. J. Srolovitz, T. J. Baker, and J.-H. Prévost. Brittle fracture in polycrystalline microstructures with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 2003; 56(14): 2015–2037. URL http://dx.doi.org/10.1002/nme.653.

[51] C. A. Duarte. *The hp Cloud Method*. Ph.D. thesis, The University of Texas at Austin, Austin, TX, USA, 1996.

[52] C. A. Duarte and J. T. Oden. An h-p adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, 1996; 139(1-4): 237–262. URL http://dx.doi.org/10.1016/S0045-7825(96)01085-7.

[53] C. A. Duarte and J. T. Oden. H-p clouds – An h-p meshless method. *Numerical Methods for Partial Differential Equations*, 1996; 12(6): 673–705. URL http://dx.doi.org/10.1002/(SICI)1098-2426(199611)12:6<673::AID-NUM3>3.0.CO;2-P.

[54] J. T. Oden, C. A. M. Duarte, and O. C. Zienkiewicz. A new cloud-based hp finite element method. *Computer Methods in Applied Mechanics and Engineering*, 1998; 153(1-2): 117–126. URL http://dx.doi.org/10.1016/S0045-7825(97)00039-X.

[55] J. M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 1996; 139(1-4): 289–314. URL http://dx.doi.org/10.1016/S0045-7825(96)01087-0.

[56] I. Babuška and J. M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 1997; 40(4): 727–758. URL http://dx.doi.org/10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N.

[57] T. Belytschko, R. Gracie, and G. Ventura. A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 2009; 17(4): 043001. URL http://stacks.iop.org/0965-0393/17/i=4/a=043001.

[58] C. A. Duarte, I. Babuška, and J. T. Oden. Generalized finite element methods for three-dimensional structural mechanics problems. *Computers & Structures*, 2000; 77(2): 215–232. URL http://dx.doi.org/10.1016/S0045-7949(99)00211-4.

[59] A. Simone, C. A. Duarte, and E. Van der Giessen. A Generalized Finite Element Method for polycrystals with discontinuous grain boundaries. *International Journal for Numerical Methods in Engineering*, 2006; 67(8): 1122–1145. URL http://dx.doi.org/10.1002/nme.1658.

[60] B. T. and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 1999; 45(5): 601–620. URL http://dx.doi.org/10.1002/(SICI)1097-0207(19990620)45:5<601::AID-NME598>3.0.CO;2-S.

[61] N. Möes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 1999; 46(1): 131–150. URL http://dx.doi.org/10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J.

[62] C. Daux, N. Moës, D. J., S. N., and B. T. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 2000; 48(12): 1741–1760. URL http://dx.doi.org/10.1002/1097-0207(20000830)48:12<1741::AID-NME956>3.0.CO;2-L.

[63] C. A. Duarte, O. N. Hamzeh, T. J. Liszka, and W. W. Tworzydlo. A generalized finite element method for the simulation of three-dimensional dynamic crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 2001; 190(15-17): 2227–2262. URL http://dx.doi.org/10.1016/S0045-7825(00)00233-4.

[64] E. Chahine, P. Laborde, and Y. Renard. A quasi-optimal convergence result for fracture mechanics with XFEM. *Comptes Rendus Mathematique*, 2006/4/1; 342(7): 527–532. URL http://dx.doi.org/10.1016/j.crma.2006.02.002.

[65] R. Merle and J. Dolbow. Solving thermal and phase change problems with the eXtended finite element method. *Computational Mechanics*, 2002; 28(5): 339–350. URL http://dx.doi.org/10.1007/s00466-002-0298-y.

[66] J. Dolbow, E. Fried, and H. Ji. Chemically induced swelling of hydrogels. *Journal of the Mechanics and Physics of Solids*, 2004/1; 52(1): 51–84. URL http://dx.doi.org/10.1016/S0022-5096(03)00091-7.

[67] J. Dolbow, E. Fried, and H. Ji. A numerical strategy for investigating the kinetic response of stimulus-responsive hydrogels. *Computer Methods in Applied Mechanics and Engineering*, 2005/10/15; 194(42-44): 4447–4480. URL http://dx.doi.org/10.1016/j.cma.2004.12.004.

[68] H. Ji, H. Mourad, E. Fried, and J. Dolbow. Kinetics of thermally induced swelling of hydrogels. *International Journal of Solids and Structures*, 2006/4; 43(7-8): 1878–1907. URL http://dx.doi.org/10.1016/j.ijsolstr.2005.03.031.

[69] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 2001; 190(32-33): 4081–4193. URL http://dx.doi.org/10.1016/S0045-7825(01)00188-8.

[70] N. Sukumar, D. L. Chopp, N. Moës, and T. Belytschko. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 2001; 190(46-47): 6183–6200. URL http://dx.doi.org/10.1016/S0045-7825(01)00215-8.

[71] N. Moës, M. Cloirec, P. Cartraud, and J. F. Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 2003; 192(28-30): 3163–3177. URL http://dx.doi.org/10.1016/S0045-7825(03)00346-3.

[72] I. Babuška and J. E. Osborn. Generalized Finite Element Methods: Their Performance and Their Relation to Mixed Methods. *SIAM Journal on Numerical Analysis*, Jun., 1983; 20(3): 510–536. URL http://www.jstor.org/stable/2157269.

[73] K. R. Srinivasan, K. Matouš, and P. H. Geubelle. Generalized finite element method for modeling nearly incompressible bimaterial hyperelastic solids. *Computer Methods in Applied Mechanics and Engineering*, 2008; 197(51-52): 4882–4893. ISSN 0045-7825. URL http://dx.doi.org/10.1016/j.cma.2008.07.014.

[74] T. Belytschko, N. Möes, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 2001; 50(4): 993–1013. URL http://dx.doi.org/10.1002/1097-0207(20010210)50:4<993::AID-NME164>3.0.CO;2-M.

[75] J. Robbins and T. E. Voth. An extended finite element formulation for modeling the response of polycrystalline materials to shock loading. In *American Physical Society, 15th APS Topical Conference on Shock Compression of Condensed Matter*. 2007; .

[76] T. P. Fries. A corrected XFEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 2008; 75(5): 503–532. URL http://dx.doi.org/10.1002/nme.2259.

[77] N. Möes, E. Béchet, and M. Tourbier. Imposing Dirichlet boundary conditions in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 2006; 67(12): 1641–1669. URL http://dx.doi.org/10.1002/nme.1675.

[78] I. Babuška, U. Banerjee, and O. J.E. Survey of meshless and generalized finite element methods: A unified approach. *Acta Numerica*, 2003; 12: 1–125. URL http://dx.doi.org/10.1017/S0962492902000090.

[79] J. Chessa, H. Wang, and T. Belytschko. On the construction of blending elements for local partition of unity enriched finite elements. *International Journal for Numerical Methods in Engineering*, 2003; 57(7): 1015–1038. URL http://dx.doi.org/10.1002/nme.777.

[80] G. Ventura, R. Gracie, and T. Belytschko. Fast integration and weight function blending in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 2009; 77(1): 1–29. URL http://dx.doi.org/10.1002/nme.2387.

[81] J. E. Tarancón, A. Vercher, E. Giner, and F. F. J. Enhanced blending elements for XFEM applied to linear elastic fracture mechanics. *International Journal for Numerical Methods in Engineering*, 2009; 77(1): 126–148. URL http://dx.doi.org/10.1002/nme.2402.

[82] R. Gracie, H. Wang, and T. Belytschko. Blending in the extended finite element method by discontinuous Galerkin and assumed strain methods. *International Journal for Numerical Methods in Engineering*, 2008; 74(11): 1645–1669.

[83] T. Strouboulis, I. Babuška, and K. Copps. The design and analysis of the Generalized Finite Element Method. *Computer Methods in Applied Mechanics and Engineering*, 2000; 181(1-3): 43–69. URL http://dx.doi.org/10.1016/S0045-7825(99)00072-9.

[84] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 1st. edition, 2002.

[85] C. A. Duarte, L. G. Reno, and A. Simone. A high-order generalized FEM for through-the-thickness branched cracks. *International Journal for Numerical Methods in Engineering*, 2007; 72(3): 325–351. URL http://dx.doi.org/10.1002/nme.2012.

[86] J. P. Pereira, C. A. Duarte, D. Guoy, and X. Jiao. hp-Generalized FEM and crack surface representation for non-planar 3-D cracks. *International Journal for Numerical Methods in Engineering*, 2009; 77(5): 601–633. URL http://dx.doi.org/10.1002/nme.2419.

[87] J. Chessa, P. Smolinski, and T. Belytschko. The extended finite element method (XFEM) for solidification problems. *International Journal for Numerical Methods in Engineering*, 2002; 53(8): 1959–1977. URL http://dx.doi.org/10.1002/nme.386.

[88] B. Vaughan, B. Smith, and D. Chopp. A comparison of the extended finite element method with the immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *Communications in Applied Mathematics and Computational Science*, 2006; 1(1): 207–228.

[89] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method: an example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering*, 2000; 47(8): 1401–1417. URL http://dx.doi.org/10.1002/(SICI)1097-0207(20000320)47:8<1401::AID-NME835>3.0.CO;2-8.

[90] W. M. Kays, M. E. Crawford, and B. Weigand. *Convective heat and mass transfer*. McGraw-Hill, 4th edition, 2004. ISBN 0072990732.

[91] C. D. Murray. The Physiological Principle of Minimum Work. I. The vascular system and cost of blood volume. *Proceedings of the National Academy of Sciences of the United States of America*, 1926; 12(3): 207–214. URL http://www.pnas.org/content/12/3/207.short.

[92] H. Hensel. Heat and Cold. *Annual Review of Physiology*, 1959; 21(1): 91–116. URL http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.ph.21.030159.000515.

[93] K. R. Morgareidge and F. N. White. Cutaneous Vascular Changes during Heating and Cooling in the Galapagos Marine Iguana. *Nature*, 1969; 223: 587–591. URL http://dx.doi.org/10.1038/223587a0.

[94] F. Seebacher and C. E. Franklin. Integration of autonomic and local mechanisms in regulating cardiovascular responses to heating and cooling in a reptile (Crocodylus porosus). *Journal of Comparative Physiology B: Biochemical, Systemic, and Environmental Physiology*, 2004; 174(7): 577–585. URL http://dx.doi.org/10.1007/s00360-004-0446-0.

[95] T. Cooper, W. C. Randall, and A. B. Hertzman. Vascular convection of heat from active muscle to overlying skin. *J Appl Physiol*, 1959; 14(2): 207–211. URL http://jap.physiology.org/cgi/content/abstract/14/2/207.

[96] G. Brengelmann. Specialized brain cooling in humans? *FASEB J.*, 1993; 7(12): 1148–1152. URL http://www.fasebj.org/cgi/content/abstract/7/12/1148.

[97] B. D. Kozola, L. A. Shipton, V. K. Natrajan, K. T. Christensen, and S. R. White. Characterization of active cooling and flow distribution in microvascular polymers. *Journal of Intelligent Material Systems and Structures*, 2010; URL http://dx.doi.org/10.1177/1045389X10379662.

[98] D. Tuckerman and R. Pease. High-performance heat sinking for VLSI. *Electron Device Letters, IEEE*, 1981; 2(5): 126–129. ISSN 0741-3106.

[99] A. Weisberg, H. H. Bau, and J. Zemel. Analysis of microchannels for integrated cooling. *International Journal of Heat and Mass Transfer*, 1992; 35(10): 2465 – 2474. ISSN 0017-9310. URL http://dx.doi.org/10.1016/0017-9310(92)90089-B.

[100] X. Wei, Y. Joshi, and M. K. Patterson. Experimental and Numerical Study of a Stacked Microchannel Heat Sink for Liquid Cooling of Microelectronic Devices. *Journal of Heat Transfer*, 2007; 129(10): 1432–1444. URL http://link.aip.org/link/?JHR/129/1432/1.

[101] P. Wu and W. A. Little. Measurement of the heat transfer characteristics of gas flow in fine channel heat exchangers used for microminiature refrigerators. *Cryogenics*, 1984; 24(8): 415 – 420. ISSN 0011-2275. URL http://www.sciencedirect.com/science/article/B6TWR-48HY7GT-XY/2/2bcd94912b65a130c352fd0bdf205f9e.

[102] A. Bejan and S. Lorente. *Design with Constructal Theory*. John Wiley & Sons, Inc., 2008. ISBN 0471998168.

[103] J. Lee, S. Lorente, and A. Bejan. Transient cooling response of smart vascular materials for self-cooling. *Journal of Applied Physics*, 2009; 105(6): 064904. URL http://link.aip.org/link/?JAP/105/064904/1.

[104] K.-M. Wang, S. Lorente, and A. Bejan. Vascular materials cooled with grids and radial channels. *International Journal of Heat and Mass Transfer*, 2009; 52(5-6): 1230 – 1239. ISSN 0017-9310. URL http://dx.doi.org/10.1016/j.ijheatmasstransfer.2008.08.027.

[105] K.-M. Wang, S. Lorente, and A. Bejan. Vascular structures for volumetric cooling and mechanical strength. *Journal of Applied Physics*, 2010; 107(4): 044901. URL http://link.aip.org/link/?JAP/107/044901/1.

[106] T. Borrvall and J. Petersson. Topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Fluids*, 2003; 41(1): 77–107. URL http://dx.doi.org/10.1002/fld.426.

[107] D. Therriault, R. Shepherd, S. White, and J. Lewis. Fugitive Inks for Direct-Write Assembly of Three-Dimensional Microvascular Networks. *Advanced Materials*, 2005; 17(4): 395–399. URL http://dx.doi.org/10.1002/adma.200400481.

[108] V. Natrajan and K. Christensen. Microscopic particle image velocimetry measurements of transition to turbulence in microscale capillaries. *Experiments in Fluids*, 2007; 43(1): 1–16. URL http://dx.doi.org/10.1007/s00348-007-0301-7.

[109] H. L. Langhaar. Steady flow in the transition length of a straight tube. *Journal of Applied Mechanics*, 1942; 9: A55–A58.

[110] V. K. Natrajan and K. T. Christensen. Two-color laser-induced fluorescent thermometry for microfluidic systems. *Measurement Science and Technology*, 2009; 20(1): 015401. URL http://dx.doi.org/10.1088/0957-0233/20/1/015401.

[111] S. Mahfoud. *Crowding and Preselection Revisited.*, volume 2 of *Parallel Problem Solving from Nature*, 27–35. Elsevier Science. ISBN 0444897305, 1992; .

[112] J. G. Siek, A. Lumsdaine, and L.-Q. Lee. *Generic programming for high performance numerical linear algebra*. Society for Industrial and Applied Mathematics. ISBN 0898714451, 1999; .

[113] T. Veldhuizen. Arrays in Blitz++. *Computing in Object-Oriented Parallel Environments*, 1998; 501–501. URL http://dx.doi.org/10.1007/3-540-49372-7_24.

[114] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic Linear Algebra Subprograms for Fortran Usage. *ACM Transactions on Mathematical Software*, 1979; 5(3): 308–323. URL http://dx.doi.org/10.1145/355841.355847.

[115] J. J. Dongarra, J. D. Croz, S. Hammarling, and I. S. Duff. A set of level 3 basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 1990; 16(1): 1–17. URL http://dx.doi.org/10.1145/77626.79170.

[116] *ISO/IEC 14882:2003 Programming languages – C++*. International Organization for Standarization.

[117] A.-W. Professional (editor). *The C++ Standard Library: A Tutorial and Reference*. Addison-Wesley Professional, 1st. edition, Nicolai M. Josuttis. ISBN 0201379260.

[118] S. Meyers. *More Effective C++: 35 New Ways to Improve Your Programs and Designs*. Addison-Wesley Professional, 1996. ISBN 020163371X.

[119] T. Veldhuizen. Expression Templates. *C++ Report*, 1995; 7(26–31).

[120] I. S. Duff, R. G. Grimes, and J. G. Lewis. Sparse matrix test problems. *ACM Trans. Math. Softw.*, 1989; 15(1): 1–14. ISSN 0098-3500. URL http://dx.doi.org/10.1145/62038.62043.

[121] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 1999; 20(3): 720–755. URL http://dx.doi.org/10.1137/S0895479895291765.

[122] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley Professional, 2001. ISBN 0201704315.

[123] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design patterns : elements of reusable object-oriented software*. Addison-Wesley Professional, 1994. ISBN 0201633612.

[124] J. O. Coplien. Curiously recurring template patterns. *C++ Report*, 1995; 7(2): 24–27. ISSN 1040-6042.

[125] J. G. Siek, L.-Q. Lee, and A. Lumsdaine. *The Boost Graph Library: User Guide and Reference Manual*. Addison-Wesley Professional, 2001. ISBN 0201729148.

Author's Biography

Alejandro Marcos Aragón Suárez was born on November 10^{th}, 1977 in San Juan, Argentina. He obtained the degree of *Ingeniero Civil* from the *Universidad Nacional de San Juan* in December of 2001, Argentina. Soon after graduation he worked as a structural engineer in *TESS*, an engineering consulting studio located in Santa Cruz de la Sierra, Bolivia. In 2004, Alejandro was awarded the prestigious Fulbright Scholarship in Argentina, to pursue a Master of Science in Civil Engineering. He obtained the M.S. in May of 2006, at the University of Illinois at Urbana-Champaign and he immediately started working on his doctoral degree, also in Civil Engineering. Under the guidance of Prof. Philippe H. Geubelle, Alejandro joined the Autonomous Materials Systems group at the University of Illinois, where his work focused on the computational design of microvascular materials. As of May of 2010, he is the first author of four publications, two of which are peer-reviewed journal publications. He also co-authored other two peer-reviewed journal publications. During 2009, he received the Mavis Memorial Scholarship Award for academic excellence and research accomplishments.